Foundry FastIron Configuration Guide

FastIron X Series Compact Switches

FastIron Edge Switch X Series

FastIron Workgroup Switch X Series

FastIron X Series Chassis

FastIron SuperX

FastIron SX 800

FastIron SX 1600

FastIron Layer 2 Compact Switches

FastIron GS

FastIron LS

FOUNDRY
NETWORKS
www.foundrynetworks.com

4980 Great America Parkway
Santa Clara, CA 95054
Tel 408.207.1600

www.foundrynet.com

March 2008

Contents

CHAPTER 1
ABOUT THIS GUIDE 1-1

CHAPTER 2
GETTING FAMILIAR WITH MANAGEMENT APPLICATIONS 2-1

CHAPTER 3
FEATURE HIGHLIGHTS 3-1

CHAPTER 4
CONFIGURING BASIC SOFTWARE FEATURES 4-1

CHAPTER 5
OPERATIONS, ADMINISTRATION, AND MAINTENANCE 5-1

CHAPTER 10
CONFIGURING METRO FEATURES 10-1

CHAPTER 11

CONFIGURING POWER OVER ETHERNET 11-1

CHAPTER 12

CONFIGURING UNI-DIRECTIONAL LINK DETECTION (UDLD) AND PROTECTED LINK GROUPS 12-1

CHAPTER 13
CONFIGURING TRUNK GROUPS AND DYNAMIC LINK AGGREGATION 13-1

Chapter 14
Configuring Virtual LANs (VLANs)....................14-1

CHAPTER 16
CONFIGURING GARP VLAN REGISTRATION PROTOCOL (GVRP)........................ 16-1

CHAPTER 17
CONFIGURING RULE-BASED IP ACCESS CONTROL LISTS (ACLS) 17-1

CHAPTER 19

CHAPTER 20

CHAPTER 21

Chapter 24
Configuring Traffic Policies 24-1

Chapter 25
Configuring LLDP and LLDP-MED.... 25-1

CHAPTER 33

CONFIGURING RIP 33-1

CHAPTER 34

CONFIGURING RIPng 34-1

CHAPTER 35

CHAPTER 36
CONFIGURING OSPF VERSION 3 (IPV6) 36-1

CHAPTER 42
CONFIGURING 802.1X PORT SECURITY 42-1

CHAPTER 43
USING THE MAC PORT SECURITY FEATURE 43-1

CHAPTER 44
CONFIGURING MULTI-DEVICE PORT AUTHENTICATION................................ 44-1

APPENDIX A

USING SYSLOG A-1

APPENDIX B

NETWORK MONITORING B-1

APPENDIX C

SOFTWARE SPECIFICATIONS C-1

Chapter 1
About This Guide

Introduction

This guide describes the following product families from Foundry Networks:

- FastIron GS and FastIron LS Layer 2 and Base Layer 3 devices
- FastIron X Series devices:
    - FastIron Edge Switch X Series (FESX) Layer 2/Layer 3 switch
    - FastIron Workgroup Switch X Series (FWSX) Layer 2 switch
    - FastIron SuperX Switch (FSX) Layer 2/Layer 3 switch
    - FastIron SX 800 and 1600 Layer 2/Layer 3 switch

This guide includes procedures for configuring the software. The software procedures show how to perform tasks using the CLI. This guide also describes how to monitor Foundry products using statistics and summary screens.

This guide applies to the following FastIron models:

- FastIron Edge Switch X Series models:
    - FastIron Edge Switch X424
    - FastIron Edge Switch X424HF
    - FastIron Edge Switch X424-POE
    - FastIron Edge Switch X624
    - FastIron Edge Switch X624E
    - FastIron Edge Switch X624HF
    - FastIron Edge Switch X624HFE
    - FastIron Edge Switch X448
    - FastIron Edge Switch X648
    - FastIron Edge Switch X648E
- FastIron GS models:
    - FastIron GS 624P
    - FastIron GS 624XGP

  - FastIron GS 648P
- FastIron LS models:
  - FastIron LS 624
  - FastIron LS 648
- FastIron SuperX Switch
- FastIron SX 800
- FastIron SX 1600
- FastIron Workgroup Switch X Series models:
  - FastIron Workgroup Switch X424
  - FastIron Workgroup Switch X448

Device Nomenclature

This guide contains the terms **FastIron Edge Switch X Series**, **FastIron SuperX Switch**, **FastIron SX**, **FastIron GS, FastIron LS**, and **FastIron WorkGroup Switch X Series**. Each term refers to a specific set of devices, as shown in Table 1.1.

Table 1.1: FastIron Family of Switches

This Name	**Refers to These Devices**
FastIron GS (FGS)	FGS624P, FGS648P, and FGS624XGP models
FastIron LS (FLS)	FLS624 and FLS648
FastIron X Series:[1]	
FastIron Edge Switch X Series (FESX)	FESX424, FESX424HF, FESX424-POE, FESX448, FESX624, , FESX624HF, FESX648
FastIron SuperX Switch (FSX) • Management Modules	FSX • 400 MHz / 256 MB • 466 MHz / 512 MB
FastIron SX • Management Modules	FSX 800 and FSX 1600 • 667 MHz / 512 MB
FastIron Workgroup Switch X Series (FWSX)	FWSX424 and FWSX448 models

[1] The FastIron X Series product family includes compact switch models and chassis models. The compact models are referred to as the FESX switches. The chassis systems are referred to as the FastIron SX switches. The chassis systems have three models: FastIron SuperX, FastIron SX 800, and FastIron SX 1600.

What’s Included in This Edition?

This edition describes the following software releases:

- For the FastIron GS products:
  - 04.2.00

  - 04.1.00
  - 04.0.00
  - 03.2.00
  - 03.1.00
  - 03.0.00
  - 02.6.00
  - 02.5.00
  - 02.4.00a
- For the FastIron LS products:
  - 04.2.00
  - 04.1.00
  - 04.0.00
  - 03.1.00 and 03.0.00
- For the FastIron Edge Switch X Series products:
  - 04.0.01 (combined FESX/FSX/FSX 800/FSX 1600/FWSX release)
  - 04.0.00 (combined FESX/FSX/FSX 800/FSX 1600/FWSX release)
  - 03.2.00 (combined FESX/FSX/FSX 800/FSX 1600/FWSX release)
  - 03.1.00 (combined FESX/FSX/FSX 800/FSX 1600/FWSX release)
  - 03.0.01a – c (combined FESX/FSX/FSX 800/FSX 1600/FWSX release)
  - 03.0.01 (combined FESX/FSX/FSX 800/FSX 1600/FWSX release)
  - 03.0.00 (combined FESX/FSX/FSX 800/FSX 1600/FWSX release)
  - 02.5.00 (combined FESX/FSX/FWSX release)
  - 02.4.00 (combined FESX/FSX/FWSX release)
  - 02.3.03 (combined FESX/FSX/FWSX release)
  - 02.3.02 (combined FESX/FSX/FWSX release)
  - 02.3.01 (combined FESX/FSX/FWSX release)
  - 02.2.00 (combined FESX/FWSX release)
  - 02.1.01
  - 02.0.00
  - 01.1.00
  - 01.0.00
- For the FastIron SuperX Switch

NOTE: Software releases for FSX devices were combined with the FESX software releases starting with FSX release 02.3.01.

  - 02.2.01
  - 02.2.00
  - 02.1.00
  - 02.0.01

- For the FastIron Workgroup Switch X Series products:

NOTE: Software releases for FWSX devices were combined with the FESX software releases starting with FESX release 02.2.00.

  - 02.0.00

What's New in this Edition

This edition describes the features in the following software release:

Summary of Enhancements in FGS Release 04.2.00

Layer 2 Enhancements in FGS 04.2.00

Feature	Description	See...
VSRP-Aware Interoperability	This enhancement introduces a new command for interoperability between FGS and FLS switches operating as VSRP-Aware devices and BigIron RX and NetIron XMR and NetIron MLX systems operating as VSRP Master devices.	"VSRP-Aware Interoperability" on page 10-38

Management Enhancements in FGS 04.2.00

Feature	Description	See...
DHCP Client-Based Auto-Configuration	DHCP Client-Based Auto-Configuration allows FGS or FLS Layer 2 and Base Layer 3 devices to automatically obtain leased IP addresses through a DHCP server, negotiate address lease renewal, and obtain a configuration file. This feature is enabled automatically.	"DHCP Client-Based Auto-Configuration" on page 29-52

Security Enhancements in FGS 04.2.00

Feature	Description	See...
AES Encryption for SSH2	With this release, Foundry SSHv2 will support the Advanced Encryption Standard (AES) data encryption methods, in addition to the Triple Data Encryption Standard (3-DES) from the previous releases.	"AES Encryption for SSHv2" on page 41-2

Audience

This guide is designed for network installers, system administrators, and resellers who will configure the software for the FastIron family of switches. This guide assumes a working knowledge of Layer 2 and Layer 3 switching and routing concepts.

If you are using Layer 3 code, you should be familiar with the following protocols if applicable to your network – IP, RIP, OSPF, BGP4, DVMRP, MBGP, IGMP, PIM, VRRP, and VRRPE.

Nomenclature

This guide uses the following typographical conventions to show information:

Italic	highlights the title of another publication and occasionally emphasizes a word or phrase.
Bold	highlights a CLI command.
Bold Italic	highlights a term that is being defined.
Underline	highlights a link on the Web management interface.
Capitals	highlights field names and buttons that appear in the Web management interface.

NOTE: A note emphasizes an important fact or calls your attention to a dependency.

WARNING: A warning calls your attention to a possible hazard that can cause injury or death.

CAUTION: A caution calls your attention to a possible hazard that can damage equipment.

Related Publications

The following Foundry Networks documents supplement the information in this guide.

- *Foundry FastIron X Series Chassis Hardware Installation Guide* – provides hardware installation procedures for the FastIron chassis devices (FSX, FSX 800 and FSX 1600).
- *Foundry FastIron Compact Switch Hardware Installation Guide* – provides hardware installation procedures for the FastIron compact switches (FES, FESX, and FWSX).
- *Foundry FastIron GS Compact Layer 2 Switch Hardware Installation Guide* – provides hardware installation procedures for the FastIron GS.
- *Foundry FastIron LS Compact Layer 2 Switch Hardware Installation Guide* – provides hardware installation procedures for the FastIron LS.
- *Foundry Management Information Base Reference* – contains the Simple Network Management Protocol (SNMP) Management Information Base (MIB) objects supported on Foundry devices.

NOTE: For the latest edition of these documents, which contain the most up-to-date information, see Product Manuals at kp.foundrynet.com.

Updates to Manuals

Manuals for this product may be updated between releases. For the latest edition of manuals, check the Foundry Knowledge Portal at kp.foundrynet.com.

How to Get Help or Report Errors

Foundry Networks is committed to ensuring that your investment in our products remains cost-effective. If you need assistance, or find errors in the manuals, contact Foundry Networks using one of the following options:

Web Access

Go to kp.foundrynet.com and log in to the Knowledge Portal (KP) to obtain more information about a product, or to report documentation errors. **To report errors, click on Cases > Create a New Ticket.**

E-mail Access

Send an e-mail to support@foundrynet.com

Telephone Access

1.877.TURBOCALL (887.2622) United States

1.408-207-1600 Outside the United States

Warranty Coverage

Contact Foundry Networks using any of the methods listed above for information about the standard and extended warranties.

Chapter 2

Getting Familiar with Management Applications

This chapter describes how to manage a Foundry device using the command line interface (CLI), the Web management interface, and IronView Network Manager software.

Logging on through the CLI

Once an IP address is assigned to a Foundry device running Layer 2 software or to an interface on the Foundry device running Layer 3 software, you can access the CLI either through the direct serial connection to the device or through a local or remote Telnet session.

You can initiate a local Telnet or SNMP connection by attaching a cable to a port and specifying the assigned management station IP address.

The commands in the CLI are organized into the following levels:

- User EXEC – Lets you display information and perform basic tasks such as pings and traceroutes.
- Privileged EXEC – Lets you use the same commands as those at the User EXEC level plus configuration commands that do not require saving the changes to the system-config file.
- CONFIG – Lets you make configuration changes to the device. To save the changes across reboots, you need to save them to the system-config file. The CONFIG level contains sub-levels for individual ports, for VLANs, for routing protocols, and other configuration areas.

NOTE: By default, any user who can open a serial or Telnet connection to the Foundry device can access all these CLI levels. To secure access, you can configure Enable passwords or local user accounts, or you can configure the device to use a RADIUS or TACACS/TACACS+ server for authentication. See the chapter “Securing Access to Management Functions” on page 39-1.

On-Line Help

To display a list of available commands or command options, enter “?” or press Tab. If you have not entered part of a command at the command prompt, all the commands supported at the current CLI level are listed. If you enter part of a command, then enter “?” or press Tab, the CLI lists the options you can enter at this point in the command string.

If you enter an invalid command followed by ?, a message appears indicating the command was unrecognized. For example:

```
FastIron(config)#rooter ip
Unrecognized command
```

Command Completion

The CLI supports command completion, so you do not need to enter the entire name of a command or option. As long as you enter enough characters of the command or option name to avoid ambiguity with other commands or options, the CLI understands what you are typing.

Scroll Control

By default, the CLI uses a page mode to paginate displays that are longer than the number of rows in your terminal emulation window. For example, if you display a list of all the commands at the global CONFIG level but your terminal emulation window does not have enough rows to display them all at once, the page mode stops the display and lists your choices for continuing the display.

Here is an example:

```
aaa
all-client
appletalk
arp
boot
```

some lines omitted for brevity...

```
ipx
lock-address
logging
mac
--More--, next page: Space, next line:
Return key, quit: Control-c
```

The software provides the following scrolling options:

- Press the Space bar to display the next page (one screen at a time).
- Press the Return or Enter key to display the next line (one line at a time).
- Press Ctrl-C or Ctrl-Q to cancel the display.

Line Editing Commands

The CLI supports the following line editing commands. To enter a line-editing command, use the CTRL-key combination for the command by pressing and holding the CTRL key, then pressing the letter associated with the command.

Table 2.1: CLI Line Editing Commands

Ctrl-Key Combination	Description
Ctrl-A	Moves to the first character on the command line.
Ctrl-B	Moves the cursor back one character.
Ctrl-C	Escapes and terminates command prompts and ongoing tasks (such as lengthy displays), and displays a fresh command prompt.
Ctrl-D	Deletes the character at the cursor.
Ctrl-E	Moves to the end of the current command line.
Ctrl-F	Moves the cursor forward one character.
Ctrl-K	Deletes all characters from the cursor to the end of the command line.

Table 2.1: CLI Line Editing Commands (Continued)

Ctrl-Key Combination	Description
Ctrl-L; Ctrl-R	Repeats the current command line on a new line.
Ctrl-N	Enters the next command line in the history buffer.
Ctrl-P	Enters the previous command line in the history buffer.
Ctrl-U; Ctrl-X	Deletes all characters from the cursor to the beginning of the command line.
Ctrl-W	Deletes the last word you typed.
Ctrl-Z	Moves from any CONFIG level of the CLI to the Privileged EXEC level; at the Privileged EXEC level, moves to the User EXEC level.

Using Slot and Port Numbers with CLI Commands

NOTE: The FastIron GS running software release 02.5.00 and later use stack, slot, and port numbers. See “Port Nomenclature on the FastIron GS and FastIron LS” on page 2-3.

Many CLI commands and displays use port numbers, or slot numbers with port numbers. The ports are labelled on the front panel of the device.

The FSX uses chassis-based port numbering which consists of a slot number and a port number. When you enter CLI commands on the FSX, you must specify both the slot number and the port number. The FESX and FWSX devices do not use this type of numbering. When you enter commands on these devices, just specify the port number. The slot numbers used in the FSX CLI examples apply only to Chassis devices.

Here is an example. The following commands change the CLI from the global CONFIG level to the configuration level for the first port on the device.

- FSX commands:

```
FastIron(config)#interface e 1/1
FastIron(config-if-1/1)#
```

- FESX and FWSX commands:

```
FastIron(config)#interface e1
FastIron(config-if-e1000-1)#
```

Port Nomenclature on the FastIron GS and FastIron LS

Platform Support:

- FGS and FLS devices running software release 02.5.00 and later

The CLI uses stack- and chassis-based port numbering. Stacking is not currently supported for the FastIron GS and FastIron LS, although the CLI supports the stack/slot/port nomenclature used with stacking technology. Full stacking port nomenclature is supported for all FastIron GS and FastIron LS devices.

Stack, Slot, and Port Numbers

Platform Support:

- FGS and FLS devices running software release 02.5.00 and later

When you enter CLI commands that include the port number as part of the syntax, you will need to specify the stack number (0 in current releases), the slot number, and the port number.

The slot and port numbers are labelled on the front of the FGS624XGP, FGS624XGP-POE, and other FastIron GS models that are shipped in accordance with the new port numbering scheme. For older FastIron GS models, an upgrade label kit is available.

FGS624P and FGS624P-POE

Figure 2.1 shows port numbers on the FGS624P and FGS624P-POE models as they appeared prior to release 02.5.00.

Figure 2.1 FGS624P and FGS624P-POE Port Numbers (Prior to Release 02.5.00)

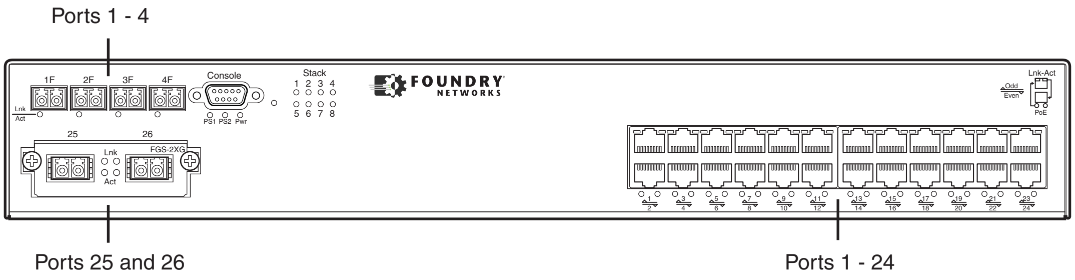


Figure 2.2 shows slot and port numbers on the FGS624P and FGS624P-POE models as they appear after release 02.5.00.

Figure 2.2 FGS624P and FGS624P-POE Slot and Port Numbers (Release 02.5.00 and later)

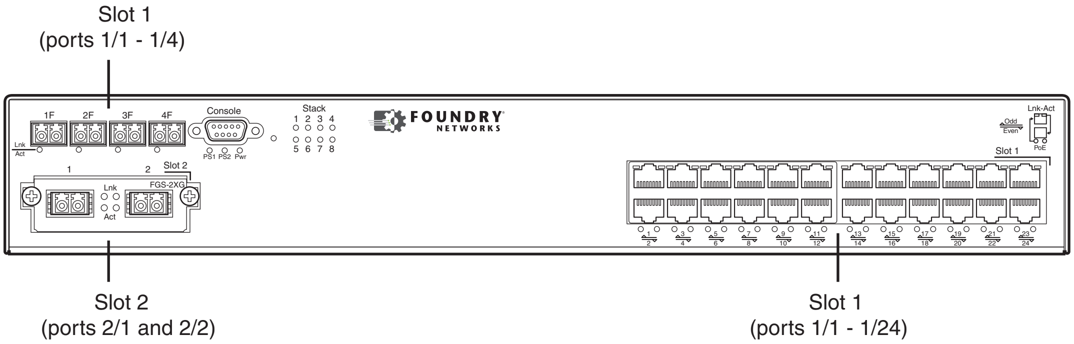


Table 2.2 shows a comparison of the old and new port numbers.

Table 2.2: Port Numbers on the FGS624P and FGS624P-POE

Port Type	Port Numbers in Pre-Release 02.5.00	Port Numbers Starting in Release 02.5.00 (Stack/Slot/Port)
GbE Copper and Fiber	1 – 24	0/1/1 – 0/1/24
10-GbE	25	0/2/1
10-GbE	26	0/2/2

FGS648P and FGS648P-POE

Figure 2.3 shows the *pre-release 02.5.00* port numbers on the FGS648P and FGS648P-POE.

Figure 2.3 FGS648P and FGS648P-POE Port Numbers (Pre-Release 02.5.00)

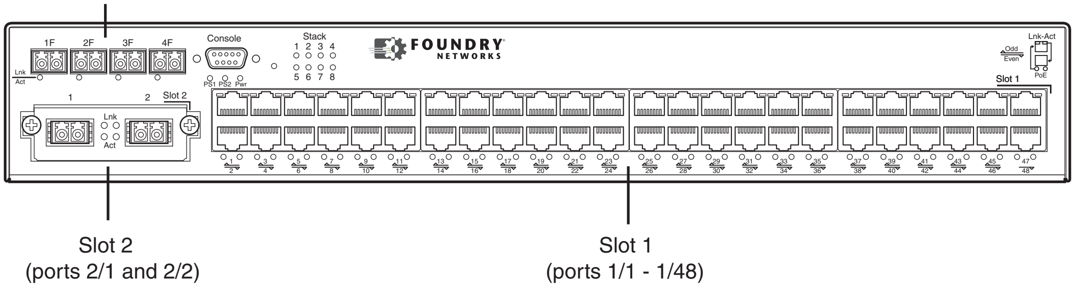


Figure 2.4 shows the new slot and port numbers in release 02.5.00 on the FGS648P and FGS648P-POE.

Figure 2.4 FGS648P and FGS648P-POE Slot and Port Numbers (Release 02.5.00 and Later)

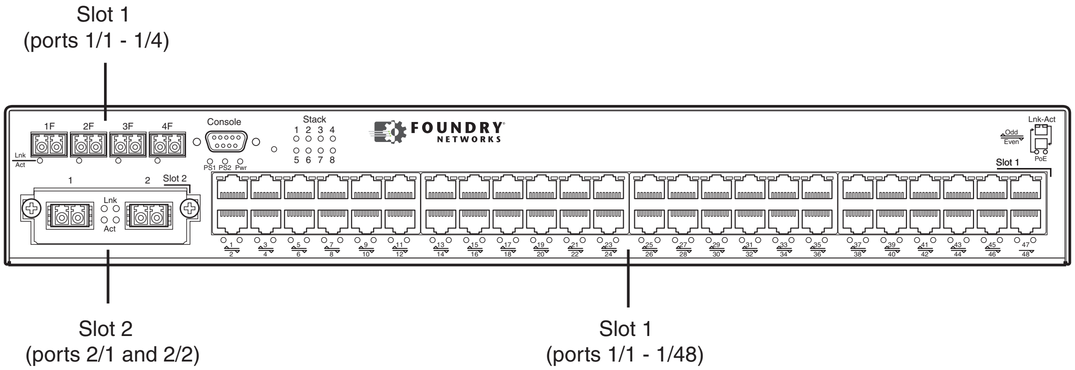


Table 2.3 shows a comparison of the old and new port numbers.

Table 2.3: Port Numbers on the FGS648P and FGS648P-POE

Port Type	Port Numbers in Pre-Release 02.5.00	Port Numbers Starting in Release 02.5.00 (Stack/Slot/Port)
GbE Copper and Fiber	1 – 48	0/1/1 – 0/1/48
10-GbE	49	0/2/1
10-GbE	50	0/2/2

FGS624XGP and FGS624XGP-POE

Figure 2.5 shows the slot and port numbers on the FGS624XGP and FGS624XGP-POE.

Figure 2.5 FGS624XGP and FGS624XGP-POE Slot and Port Numbers

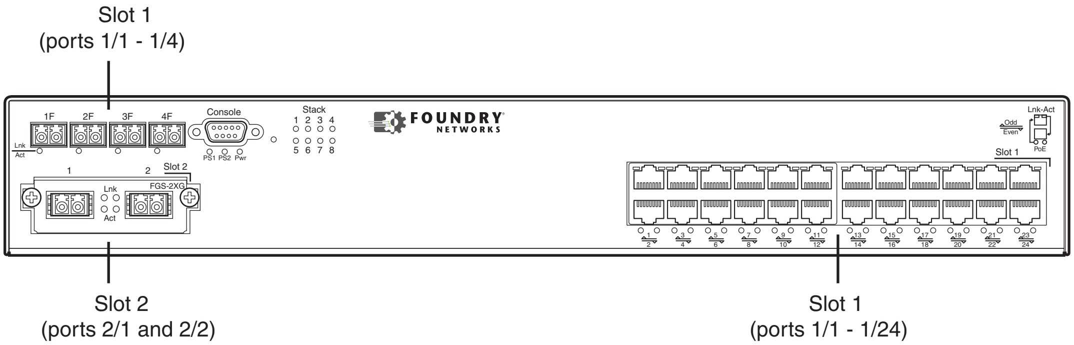


Table 2.4 shows the port numbers.

Table 2.4: Port Numbers on the FGS624XGP and FGS624XGP-POE-POE

Port Type	Port Numbers (Stack/Slot/Port)
GbE Copper and Fiber	0/1/1 – 0/1/48
2-port 10-GbE (left-most port)	0/2/1
2-port 10-GbE (right-most-port)	0/2/2
1-port 10-GbE	0/3/1

Using the FastIron GS Port Nomenclature

Platform Support:

- FGS and FLS devices running software release 02.5.00 and later

When you enter a CLI command that includes the port number as part of the syntax, you must use the stack/slot/port number format. For example, the following commands change the CLI from the global CONFIG level to the configuration level for the first port on the device.

- FastIron GS commands prior to release 02.5.00:

```
FGS624(config)#interface e 1
FGS624(config-if-e1000-1)#
```

- FastIron GS commands starting in release 02.5.00:

```
FGS624(config)#interface e 0/1/1
FGS624(config-if-e1000-0/1/1)#
```

Syntax: ethernet <stacknum>/<slotnum>/<portnum>

The <stacknum> parameter is required on FastIron GS devices running software release 02.5.00 or later.

The <slotnum> parameter is required on chassis devices and on FastIron GS devices running software release 02.5.00 or later.

The <portnum> parameter is a valid port number.

Searching and Filtering Output from CLI Commands

You can filter CLI output from **show** commands and at the --More-- prompt. You can search for individual characters, strings, or construct complex regular expressions to filter the output.

Searching and Filtering Output from Show Commands

You can filter output from **show** commands to display lines containing a specified string, lines that do not contain a specified string, or output starting with a line containing a specified string. The search string is a regular expression consisting of a single character or string of characters. You can use special characters to construct complex regular expressions. See “Using Special Characters in Regular Expressions” on page 2-9 for information on special characters used with regular expressions.

Displaying Lines Containing a Specified String

The following command filters the output of the **show interface** command for port 3/11 so it displays only lines containing the word “Internet”. This command can be used to display the IP address of the interface.

```
FastIron#show interface e 3/11 | include Internet
  Internet address is 192.168.1.11/24, MTU 1518 bytes, encapsulation ethernet
```

Syntax: <show-command> | include <regular-expression>

NOTE: The vertical bar (|) is part of the command.

Note that the regular expression specified as the search string is case sensitive. In the example above, a search string of “Internet” would match the line containing the IP address, but a search string of “internet” would not.

Displaying Lines That Do Not Contain a Specified String

The following command filters the output of the **show who** command so it displays only lines that do not contain the word “closed”. This command can be used to display open connections to the Foundry device.

```
FastIron#show who | exclude closed
Console connections:
        established
        you are connecting to this session
        2 seconds in idle
Telnet connections (inbound):
 1      established, client ip address 192.168.9.37
        27 seconds in idle
Telnet connection (outbound):
SSH connections:
```

Syntax: <show-command> | exclude <regular-expression>

Displaying Lines Starting with a Specified String

The following command filters the output of the **show who** command so it displays output starting with the first line that contains the word “SSH”. This command can be used to display information about SSH connections to the Foundry device.

```
FastIron#show who | begin SSH
SSH connections:
 1      established, client ip address 192.168.9.210
        7 seconds in idle
 2      closed
 3      closed
 4      closed
 5      closed
```

Syntax: <show-command> | begin <regular-expression>

Searching and Filtering Output at the --More-- Prompt

The --More-- prompt displays when output extends beyond a single page. From this prompt, you can press the Space bar to display the next page, the Return or Enter key to display the next line, or Ctrl-C or Q to cancel the display. In addition, you can search and filter output from this prompt.

At the --More-- prompt, you can press the forward slash key (/) and then enter a search string. The Foundry device displays output starting from the first line that contains the search string, similar to the **begin** option for **show** commands. For example:

```
--More--, next page: Space, next line: Return key, quit: Control-c
/telnet
```

The results of the search are displayed:

```
searching...
  telnet              Telnet by name or IP address
  temperature         temperature sensor commands
  terminal            display syslog
  traceroute          TraceRoute to IP node
  undebug             Disable debugging functions (see also 'debug')
  undelete            Undelete flash card files
  whois               WHOIS lookup
  write               Write running configuration to flash or terminal
```

To display lines containing only a specified search string (similar to the **include** option for **show** commands) press the plus sign key (+) at the --More-- prompt and then enter the search string.

```
--More--, next page: Space, next line: Return key, quit: Control-c
+telnet
```

The filtered results are displayed:

```
filtering...
  telnet              Telnet by name or IP address
```

To display lines that do not contain a specified search string (similar to the **exclude** option for **show** commands) press the minus sign key (-) at the --More-- prompt and then enter the search string.

```
--More--, next page: Space, next line: Return key, quit: Control-c
-telnet
```

The filtered results are displayed:

```
filtering...
  temperature         temperature sensor commands
  terminal            display syslog
  traceroute          TraceRoute to IP node
  undebug             Disable debugging functions (see also 'debug')
  undelete            Undelete flash card files
  whois               WHOIS lookup
  write               Write running configuration to flash or terminal
```

As with the commands for filtering output from **show** commands, the search string is a regular expression consisting of a single character or string of characters. You can use special characters to construct complex regular expressions. See the next section for information on special characters used with regular expressions.

Using Special Characters in Regular Expressions

You use a regular expression to specify a single character or multiple characters as a search string. In addition, you can include special characters that influence the way the software matches the output against the search string. These special characters are listed in the following table.

Table 2.5: Special Characters for Regular Expressions

Character	Operation
.	The period matches on any single character, including a blank space. For example, the following regular expression matches “aaz”, “abz”, “acz”, and so on, but not just “az”: a.z
*	The asterisk matches on zero or more sequential instances of a pattern. For example, the following regular expression matches output that contains the string “abc”, followed by zero or more Xs: abcX*
+	The plus sign matches on one or more sequential instances of a pattern. For example, the following regular expression matches output that contains "de", followed by a sequence of “g”s, such as “deg”, “degg”, “deggg”, and so on: deg+

Table 2.5: Special Characters for Regular Expressions (Continued)

Character	Operation
?	The question mark matches on zero occurrences or one occurrence of a pattern. For example, the following regular expression matches output that contains "dg" or "deg": de?g **Note:** Normally when you type a question mark, the CLI lists the commands or options at that CLI level that begin with the character or string you entered. However, if you enter Ctrl-V and then type a question mark, the question mark is inserted into the command line, allowing you to use it as part of a regular expression.
^	A caret (when not used within brackets) matches on the beginning of an input string. For example, the following regular expression matches output that begins with “deg”: ^deg
$	A dollar sign matches on the end of an input string. For example, the following regular expression matches output that ends with “deg”: deg$
_	An underscore matches on one or more of the following: • **,** (comma) • **{** (left curly brace) • **}** (right curly brace) • **(** (left parenthesis) • **)** (right parenthesis) • The beginning of the input string • The end of the input string • A blank space For example, the following regular expression matches on “100” but not on “1002”, “2100”, and so on. _100_
[ ]	Square brackets enclose a range of single-character patterns. For example, the following regular expression matches output that contains “1”, “2”, “3”, “4”, or “5”: [1-5] You can use the following expression symbols within the brackets. These symbols are allowed only inside the brackets. • **^ –** The caret matches on any characters ***except*** the ones in the brackets. For example, the following regular expression matches output that does ***not*** contain “1”, “2”, “3”, “4”, or “5”: [^1-5] • **-** The hyphen separates the beginning and ending of a range of characters. A match occurs if any of the characters within the range is present. See the example above.

Table 2.5: Special Characters for Regular Expressions (Continued)

Character	Operation
|	A vertical bar separates two alternative values or sets of values. The output can match one or the other value. For example, the following regular expression matches output that contains either "abc" or "defg": abc|defg
( )	Parentheses allow you to create complex expressions. For example, the following complex expression matches on "abc", "abcabc", or "defg", but not on "abcdefgdefg": ((abc)+)|((defg)?)

If you want to filter for a special character instead of using the special character as described in the table above, enter "\" (backslash) in front of the character. For example, to filter on output containing an asterisk, enter the asterisk portion of the regular expression as "*".

```
FastIron#show ip route bgp | include \*
```

Creating an Alias for a CLI Command

Platform Support:

- FGS and FLS devices running software release 03.0.00 and later
- FESX/FSX/FWSX devices running software release 03.2.00 and later

You can create ***aliases*** for CLI commands. An alias serves as a shorthand version of a longer CLI command. For example, you can create an alias called **shoro** for the CLI command **show ip route**. Then when you enter **shoro** at the command prompt, the **show ip route** command is executed.

To create an alias called **shoro** for the CLI command **show ip route**, enter the following command:

```
FastIron(config)#alias shoro = show ip route
```

Syntax: [no] alias <alias-name> = <cli-command>

The <alias-name> must be a single word, without spaces.

After the alias is configured, entering **shoro** at either the Privileged EXEC or CONFIG levels of the CLI, executes the **show ip route** command.

To create an alias called **wrsbc** for the CLI command **copy running-config tftp 10.10.10.10 test.cfg**, enter the following command:

```
FastIron(config)#alias wrsbc = copy running-config tftp 10.10.10.10 test.cfg
```

To remove the **wrsbc** alias from the Foundry device's configuration, enter one of the following commands:

```
FastIron(config)#no alias wrsbc
```

or

```
FastIron(config)#unalias wrsbc
```

Syntax: unalias <alias-name>

The specified <alias-name> must be the name of an alias already configured on the Foundry device.

To display the aliases currently configured on the Foundry device, enter the following command at either the Privileged EXEC or CONFIG levels of the CLI:

```
FastIron#alias
```

```
          wrsbc      copy running-config tftp 10.10.10.10 test.cfg
           shoro       show ip route
```

Syntax: alias

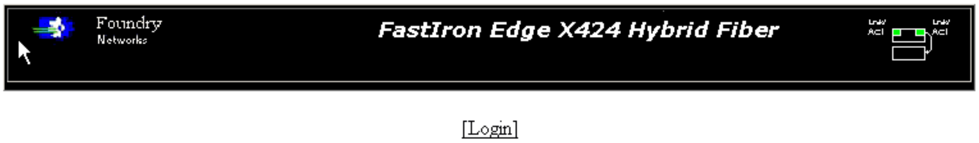
Configuration Notes

- You cannot include additional parameters with the alias at the command prompt. For example, after you create the **shoro** alias, **shoro bgp** would not be a valid command.
- If configured on the Foundry device, authentication, authorization, and accounting is performed on the actual command, not on the alias for the command.
- To save an alias definition to the device's startup-config file, use the **write memory** command.

Logging On through the Web Management Interface

To use the Web management interface, open a web browser and enter the IP address of the Foundry device's management port in the Location or Address field. The web browser contacts the Foundry device and displays a Login panel, such as the one shown below for the FESX.

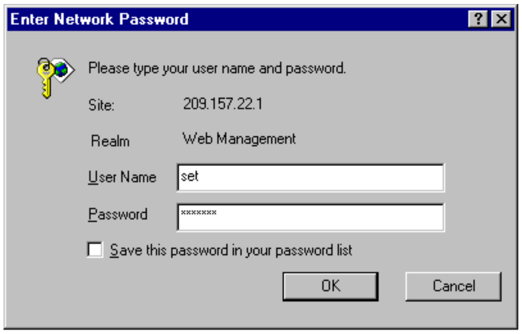
Figure 2.6 Web Management Interface Login Panel



NOTE: If you are unable to connect with the device through a Web browser due to a proxy problem, it may be necessary to set your Web browser to direct Internet access instead of using a proxy. For information on how to change a proxy setting, refer to the on-line help provided with your Web browser.

To log in, click on the Login link. The following dialog box is displayed.

Figure 2.7 Web Management Interface Login Dialog


Enter Network Password

Please type your user name and password.

Site: 209.157.22.1

Realm Web Management

User Name set

Password *******

☐ Save this password in your password list

The login username and password you enter depends on whether your device is configured with AAA authentication for SNMP. If AAA authentication for SNMP is not configured, you can use the user name "get" and the default read-only password "public" for read-only access. However, for read-write access, you must enter "set" for the user name, and enter a read-write community string you have configured on the device for the password. There is no default read-write community string. You must add one using the CLI.

As an alternative to using the SNMP community strings to log in, you can configure the Foundry device to secure Web management access using local user accounts or Access Control Lists (ACLs).

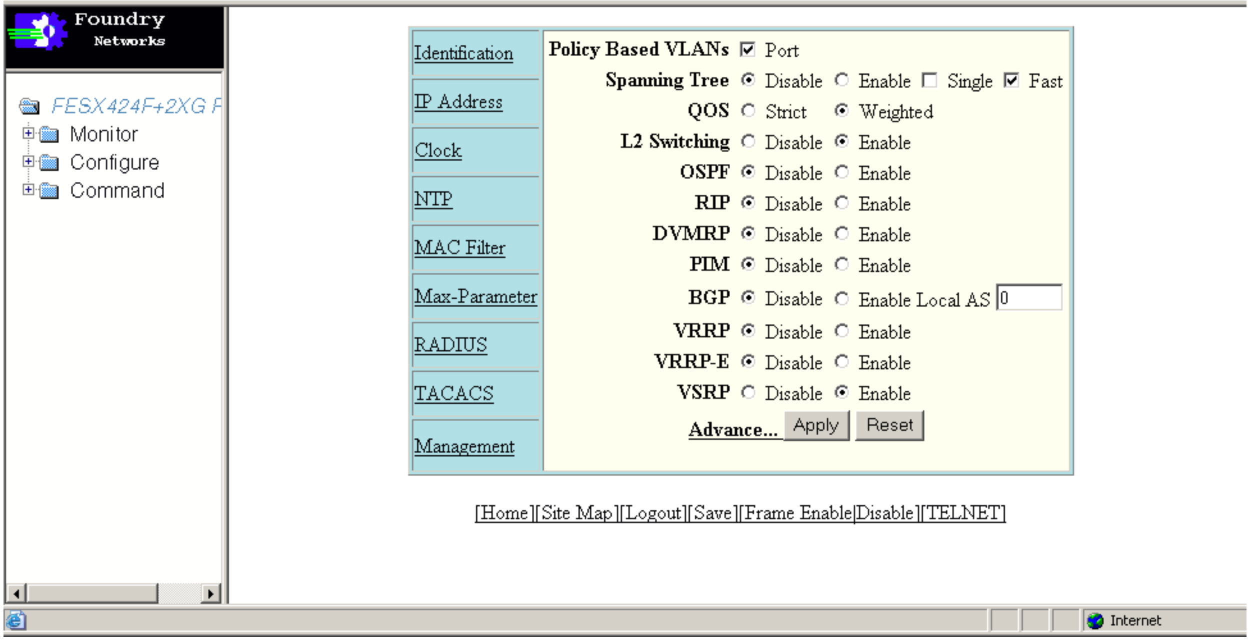
Navigating the Web Management Interface

When you log into a device, the System configuration panel is displayed. This panel allows you to enable or disable major system features. You can return to this panel from any other panel by selecting the Home link.

The Site Map link gives you a view of all available options on a single screen.

Figure 2.8 displays the first Web management interface panel for Layer 3 Switch features, while Figure 2.9 displays the first panel for Layer 2 Switch features. These panels allow you to configure the features supported by the Layer 3 Switch and Layer 2 Switch software.

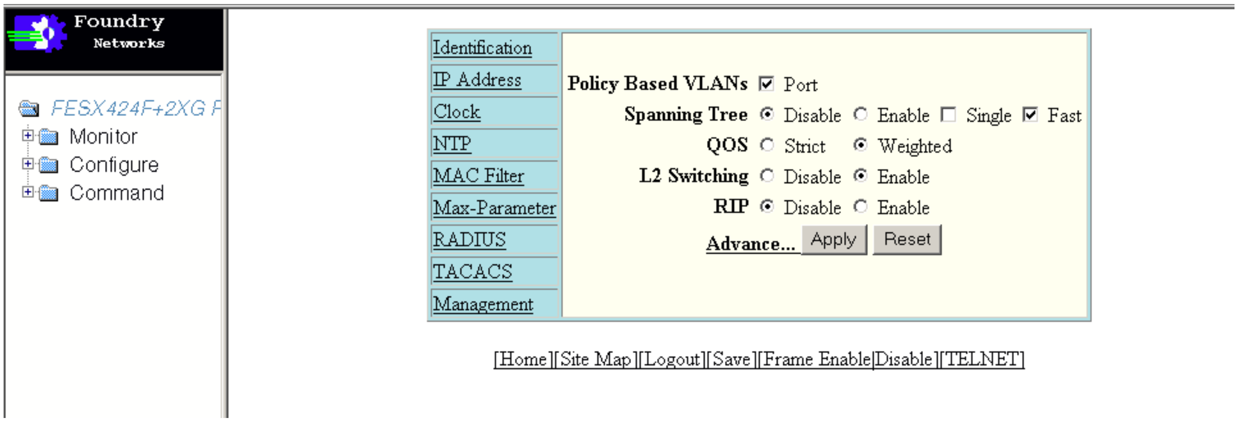
Figure 2.8 First Panel for Layer 3 Switch Features



Policy Based VLANs ☑ Port
Spanning Tree ● Disable ○ Enable ☐ Single ☑ Fast
QOS ○ Strict ● Weighted
L2 Switching ○ Disable ● Enable
OSPF ● Disable ○ Enable
RIP ● Disable ○ Enable
DVMRP ● Disable ○ Enable
PIM ● Disable ○ Enable
BGP ● Disable ○ Enable Local AS 0
VRRP ● Disable ○ Enable
VRRP-E ● Disable ○ Enable
VSRP ○ Disable ● Enable

NOTE: If you are using Internet Explorer 6.0 to view the Web management interface, make sure the version you are running includes the latest service pack(s). Otherwise, the navigation tree (the left-most pane in Figure 2.8) will not display properly. For information on how to load the latest service pack(s), refer to the on-line help provided with your Web browser.

Figure 2.9 First Panel for Layer 2 Switch Features



Policy Based VLANs ☑ Port
Spanning Tree ● Disable ○ Enable ☐ Single ☑ Fast
QOS ○ Strict ● Weighted
L2 Switching ○ Disable ● Enable
RIP ● Disable ○ Enable

NOTE: If you are using Internet Explorer 6.0 to view the Web management interface, make sure the version you are running includes the latest service pack(s). Otherwise, the navigation tree (the left-most pane in Figure 2.8) will not display properly. For information on how to load the latest service pack(s), refer to the on-line help provided with your Web browser.

The left pane of the Web management interface window contains a “tree view,” similar to the one found in Windows Explorer. Configuration options are grouped into folders in the tree view. These folders, when expanded, reveal additional options. To expand a folder, click on the plus sign to the left of the folder icon.

You can configure the appearance of the Web management interface by using one of the following methods.

Using the CLI, you can modify the appearance of the Web management interface with the **web-management** command.

To cause the Web management interface to display the List view by default:

```
FastIron(config)#web-management list-menu
```

To disable the front panel frame:

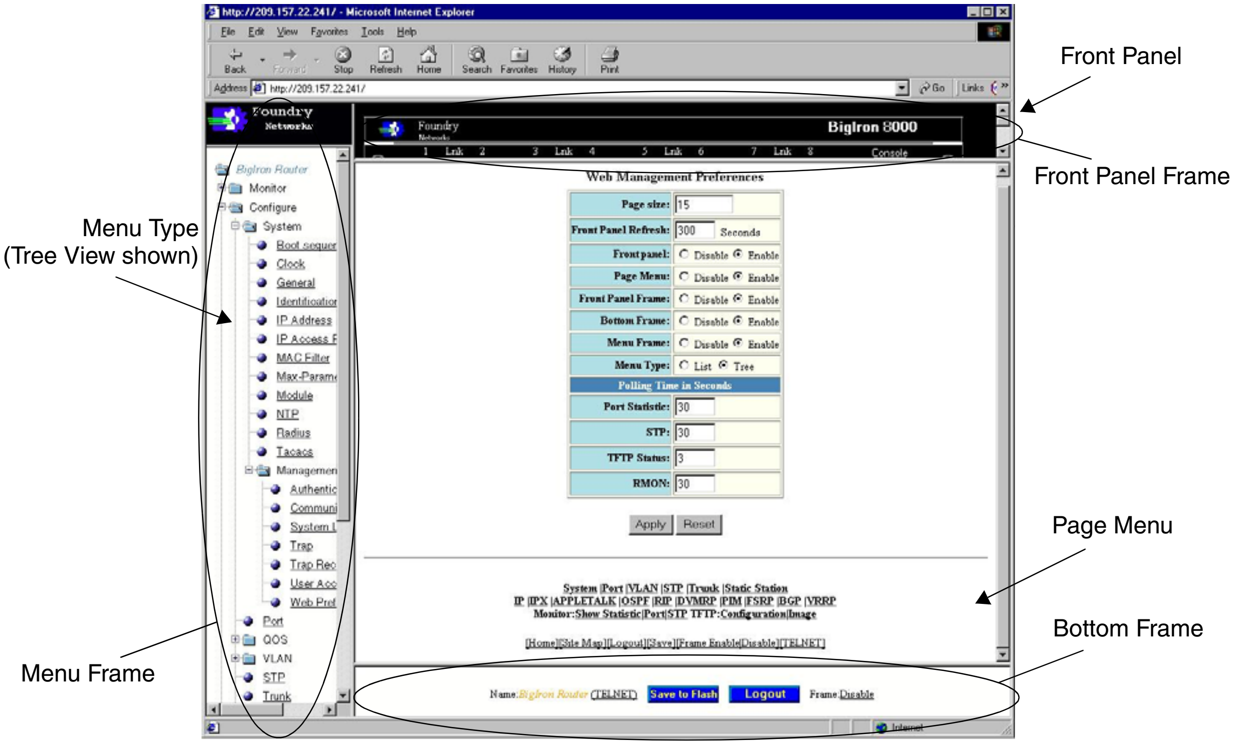
```
FastIron(config)#no web-management front-panel
```

When you save the configuration with the **write memory** command, the changes will take place the next time you start the Web management interface, or if you are currently running the Web management interface, the changes will take place when you click the Refresh button on your browser.

USING THE WEB MANAGEMENT INTERFACE

1. Click on the plus sign next to Configure in the tree view to expand the list of configuration options.
2. Click on the plus sign next to System in the tree view to expand the list of system configuration links.
3. Click on the plus sign next to Management in the tree view to expand the list of system management links.
4. Click on the Web Preference link to display the Web Management Preferences panel.

5. Enable or disable elements on the Web management interface by clicking on the appropriate radio buttons on the panel. The following figure identifies the elements you can change.



Web Management Preferences

Page size:	15
Front Panel Refresh:	300 Seconds
Front panel:	○ Disable ● Enable
Page Menu:	○ Disable ● Enable
Front Panel Frame:	○ Disable ● Enable
Bottom Frame:	○ Disable ● Enable
Menu Frame:	○ Disable ● Enable
Menu Type:	○ List ● Tree
Polling Time in Seconds	
Port Statistic:	30
STP:	30
TFTP Status:	3
RMON:	30

System |Port |VLAN |STP |Trunk |Static Station
IP |IPX |APPLETALK |OSPF |RIP |DVMRP |PIM |FSRP |BGP |VRRP
Monitor:Show Statistic|Port|STP TFTP:Configuration|Image

[Home][Site Map][Logout][Save][Frame Enable|Disable][TELNET]

Name:BigIron Router (TELNET) Save to Flash Logout Frame:Disable

---

NOTE: The tree view is available when you use the Web management interface with Netscape 4.0 or higher or Internet Explorer 4.0 or higher browsers. If you use the Web management interface with an older browser, the Web management interface displays the List view only, and the Web Management Preferences panel does not include an option to display the tree view.

---

6. When you have finished, click the Apply button on the panel, then click the Refresh button on your browser to activate the changes.
7. To save the configuration, click the plus sign next to the Command folder, then click the Save to Flash link.

NOTE: The only changes that become permanent are the settings to the Menu Type and the Front Panel Frame. Any other elements you enable or disable will go back to their default settings the next time you start the Web management interface.

Logging on Through IronView Network Manager

See the *Foundry IronView® Network Manager - IronPoint Edition User Guide* for information about using IronView Network Manager.

Chapter 3

Feature Highlights

FESX, FSX, FSX 800, and FSX 1600 devices support many of the applicable system-level, Layer 2, and Layer 3 features supported on the BigIron chassis devices. FGS and FLS devices support system-level, Layer 2 and Base Layer 3 features. FWSX devices support system-level and Layer 2 features only. The features that are available depend on the type of software image the device is running. You can run one of the following types of software images on these devices:

- Layer 2 (supported on all models)
- Base Layer 3 (supported on the FESX, FSX, FSX 800, FSX 1600, FGS and FLS)
- Full Layer 3 (supported on FESX, FSX, FSX 800, and FSX 1600 premium models only)

This chapter contains information under the following headings:

- "Supported Features" on page 3-2
- "Unsupported Features" on page 3-17
- "Supported IPv6 Management Features" on page 3-18

Table 3.1 lists the software that is loaded into the device's primary and secondary flash areas at the factory. All the flash images are included on the CD-ROM shipped with the device.

Table 3.1: Default Software Loads

Model	Software Images	
	Primary Flash	Secondary Flash
All FESX, FSX, FSX 800, and FSX 1600 non-premium models	Layer 2	Base Layer 3
All FESX, FSX, FSX 800, and FSX 1600 premium models	Full Layer 3	Layer 2
All FWSX models	Layer 2	Layer 2
All FGS models	Layer 2	Base Layer 3
All FLS models	Layer 2	Base Layer 3

Supported Features

Table 3.2 lists the feature highlights in the FastIron X Series, FLS, and FGS software.

Supported Management Features

Table 3.2 lists the management features that are supported for the FastIron platforms.

Table 3.2: Supported Management Features

Category, Description, and Configuration Notes	Supported on	
	FESX, FSX, FSX 800, FSX 1600, FWSX	**FGS, FLS**
AAA support for console commands	X	
Access Control Lists (ACLs) for controlling management access	X	X
DHCP Client-Based Auto-Configuration		X
Combined DSCP and internal marking in one ACL rule	X	
DSCP Mapping for values 1 through 8	X	**X**
Configuring an interface as the source for all TFTP, Syslog, and SNTP packets	X	
Disabling TFTP Access	X	
IPv6 management	X	X
IronView Network Manager (optional standalone and HP OpenView GUI)	X	X
P-Bridge and Q-Bridge MIBs - RFC 2674	X	X
Port flap dampening	X	X
Remote monitoring (RMON)	X	X
Retaining Syslog messages after a soft reboot	X	
sFlow • RFC 3176 • For inbound traffic only • 802.1X username export support for encrypted and non-encrypted EAP types	X	X
Serial and Telnet access to industry-standard Command Line Interface (CLI)	X	X
Show log on all terminals	X	X
SNMP v1, v2, v3	X	X
SNMP V3 traps	X	X
SNTP over IPv6	X	X

Table 3.2: Supported Management Features (Continued)

Category, Description, and Configuration Notes	Supported on	
	FESX, FSX, FSX 800, FSX 1600, FWSX	FGS, FLS
Specifying the maximum number of entries allowed in the RMON Control Table	X	
Specifying which IP address will be included in a DHCP/BOOTP reply packet	X	
Traffic counters for outbound traffic	X	
Web-based GUI	X	X

Supported Security Features

Table 3.3 lists the supported security features for FastIron products.

Table 3.3: Supported Security Features

Category, Description, and Configuration Notes	Supported on	
	FESX, FSX, FSX 800, FSX 1600, FWSX	**FGS, FLS**
802.1X port security	X	X
Access Control Lists (ACLs) for filtering transit traffic • Support for inbound ACLs only. These devices do not support outbound ACLs.	X	X
Address locking	X	X
AES Encryption for SSH v2	X	X
Authentication, Authorization and Accounting (AAA) • RADIUS, TACACS/TACACS+	X	X
Denial of Service (DoS) protection • TCP SYN Attacks and ICMP Attacks	X	X
DHCP Snooping	X	
802.1X dynamic assignment for ACL, MAC filter, and VLAN	X	X
Dynamic ACLs with Multi-Device Port Authentication	X	X
Dynamic ARP Inspection	X	
EAP Pass-through Support	X	X
Enhancements to username and password	X	X
HTTPS	X	X
IP Source Guard	X	
Layer 2 MAC filtering • Filtering on source and destination MAC addresses	X	X
Local passwords	X	X
MAC authentication password override	X	X
MAC filter override of 802.1X	X	X
Ability to disable MAC Learning	X	X
MAC port security	X	X
Multi-device port authentication	X	X
Multiple-device port authentication with dynamic VLAN assignment	X	X

Table 3.3: Supported Security Features (Continued)

Category, Description, and Configuration Notes	Supported on	
	FESX, FSX, FSX 800, FSX 1600, FWSX	FGS, FLS
MAC authentication RADIUS timeout action	X	X
802.1X authentication RADIUS timeout action	X	X
Secure Copy (SCP)	X	X
Secure Shell (SSH) v2	X	X
Packet filtering on TCP Flags		X

Supported System Level Features

Table 3.4 lists the supported system features for FastIron products.

Table 3.4: Supported System Level Features

Category, Description, and Configuration Notes	Supported on	
	FESX, FSX, FSX 800, FSX 1600, FWSX	FGS, FLS
10/100/1000 port speed	X	X
16,000 MAC addresses per switch	X	X
802.3ad link aggregation (dynamic trunk groups) • Foundry ports enabled for link aggregation follow the same rules as ports configured for trunk groups. See “Trunk Group Rules” on page 13-3.	X	X
ACL-based mirroring	X	X
ACL-based rate limiting • The FastIron X Series devices support ACL-based fixed and adaptive rate limiting on inbound ports • The FGS and FLS devices support ACL-based fixed rate limiting on inbound ports	X	X
ACL filtering based on VLAN membership or VE port membership	X	
ACL logging of denied packets • ACL logging is supported for denied packets, which are sent to the CPU for logging • ACL logging is not supported for permitted packets • Packets that are denied by ACL filters are logged in the Syslog based on a sample time-period.	X	
ACL statistics	X	
ACLs to filter ARP packets	X	
Alias Command	X	X
Asymmetric flow control • Responds to flow control packets, but does not generate them.	X	X
Auto MDI/MDIX	X	X
Auto-negotiation	X	X
Automatic removal of Dynamic VLAN for 802.1X ports		X
Broadcast, multicast, and unknown-unicast rate limiting	X	X
Boot and reload after 5 minutes at or above shutdown temperature	X	X

Table 3.4: Supported System Level Features (Continued)

Category, Description, and Configuration Notes	Supported on	
	FESX, FSX, FSX 800, FSX 1600, FWSX	FGS, FLS
DiffServ support	X	X
Digital optical monitoring	X	X
Displaying interface names in Syslog	X	X
Displaying TCP/UDP port numbers in Syslog messages	X	X
Dynamic buffer allocation	X	X
Fixed rate limiting • The FESX, FSX, FSX 800, FSX 1600, and FWSX, and FESX6 support: • Port-based rate limiting on inbound ports • Fixed rate limiting is not supported on 10-Gigabit Ethernet ports. • Fixed rate limiting is not supported on tagged ports in the full Layer 3 router image • The FGS supports: • Port-based fixed rate limiting on inbound ports • Port-based and port- and priority-based rate limiting on outbound ports • The above are supported on Gigabit and 10-Gigabit Ethernet ports.	X	X
Foundry Discovery Protocol (FDP) / Cisco Discovery Protocol (CDP)	X	X
Generic buffer profile		X
Multiple Syslog server logging • Up to six Syslog servers	X	X
Negative temperature setting	X	X
OSPF Version 2 MIB • RFC 1850	X	
Outbound rate shaping • The FGS does not support outbound rate shaping. It supports *outbound rate limiting*	X	
Path MTU Discovery (RFC 1191) support • Note: IP MTU is NOT supported for FGS and FLS	X	
Port mirroring and monitoring • Mirroring of both inbound and outbound traffic on individual ports is supported.	X	X

Table 3.4: Supported System Level Features (Continued)

Category, Description, and Configuration Notes	Supported on	
	FESX, FSX, FSX 800, FSX 1600, FWSX	FGS, FLS
Power over Ethernet	X	X (FGS only)
Priority mapping using ACLs	X	X
Protected link groups • Supported on GbE ports only. Not supported on 10-GbE ports.	X	X
Specifying a Simple Network Time Protocol (SNTP) Server	X	X
Specifying the minimum number of ports in a trunk group	X	X
Static MAC entries with option to set priority	X	X
Static Multicast MAC	X	X

Supported Layer 2 Features

Table 3.5 lists the supported Layer 2 features for FastIron products.

Table 3.5: Supported Layer 2 Features

Category, Description, and Configuration Notes	Supported on	
	FESX, FSX, FSX 800, FSX 1600, FWSX	**FGS, FLS**
802.1D Spanning Tree Support • Enhanced IronSpan support includes Fast Port Span and Single-instance Span • Foundry Layer 2 devices (switches) support up to 254 spanning tree instances for VLANs. • Foundry Layer 3 devices (routers) support up to 254 spanning tree instances for VLANs. • PVST/PVST+ compatibility • PVRST compatibility	X	X
802.1p Quality of Service (QoS) • Strict Priority (SP). • Weighted Round Robin (WRR) • Combined SP and WRR • 8 priority queues	X	X
802.1s Multiple Spanning Tree	X	X
802.1W Rapid Spanning Tree (RSTP) • 802.1W RSTP support allows for sub-second convergence (both final standard and draft 3 supported)	X	X
PVRST compatibility	X	X
ACL-based rate limiting QoS	X	X
ACLs	X	X
BPDU Guard	X	X
Root Guard	X	X
Configuring Uplink Ports Within a Port-Based VLAN	X	X
Dynamic Host Configuration Protocol (DHCP) Assist	X	X
Extended MRP ring IDs from 1 – 1023	X	
IGMP v1/v2 Snooping Global	X	X
IGMP v3 Snooping Global	X (*,G)	X (S,G)
IGMP v1/v2/v3 Snooping per VLAN	X	X

Table 3.5: Supported Layer 2 Features (Continued)

Category, Description, and Configuration Notes	Supported on	
	FESX, FSX, FSX 800, FSX 1600, FWSX	FGS, FLS
IGMP v2/v3 Fast Leave	X	X
IGMP Tracking	X	X
IGMP Filters	X	X
Interpacket Gap (IPG) adjustment	X	X
Jumbo frames • 1-Gigabit and 10-Gigabit Ethernet ports • Up to 9216 bytes	X	X
Jumbo frames 10/100 support (up to 10240 bytes)	X	X
Link Fault Signaling (LFS) for 10-Gigabit Ethernet ports only	X	X
LLDP and LLDP-MED	X	X
MAC-Based VLANs • Dynamic MAC-Based VLAN Activation		X
Metro Ring Protocol 1 (MRP 1)	X	X
Metro Ring Protocol 2 (MRP 2)	X	
MLD Snooping V1/V2 • MLD V1/V2 snooping (global and local) • MLD fast leave for V1 • MLD tracking and fast leave for V2 • Static MLD and IGMP groups with support for proxy	X	X
PIM-SM V2 Snooping	X	X
Remote Fault Notification (RFN) for Gigabit Ethernet ports	X	X
LACP • LACP trunk group ports follow the same configuration rules as for statically configured trunk group ports. • Support for single link LACP	X	X
Trunk groups • Option to include L2 in trunk hash calculation • Support for trunk threshold	X	X
Flexible trunk group membership		X
Topology groups	X	X
Uni-directional Link Detection (UDLD) (Link keepalive)	X	X

Table 3.5: Supported Layer 2 Features (Continued)

Category, Description, and Configuration Notes	Supported on	
	FESX, FSX, FSX 800, FSX 1600, FWSX	**FGS, FLS**
Virtual Cable Testing (VCT) technology • Uses Time Domain Reflectometry (TDR) technology to detect and report cable statistics such as; local and remote link pair, cable length, and link status.	X	X
Virtual Switch Redundancy Protocol (VSRP)	X	X
VSRP-Aware security features	X	X
VLAN Support: • 4096 maximum VLANs • 802.1Q with tagging • 802.1Q-in-Q Super Aggregated VLANs (SAVs) • Dual-mode VLANs • GVRP • Layer 2 VLANS (untagged ports only) • Protocol VLANs (AppleTalk, IPv4, dynamic IPv6, and IPX) • Layer 3 Subnet VLANs (Appletalk, IP subnet network, and IPX) • VLAN groups	X	X
VLAN-based mirroring		X
VSRP and MRP signaling	X	X
VSRP Fast Start	X	X

Supported Base Layer 3 Features

Table 3.6 lists the supported Base Layer 3 features for FastIron products.

Table 3.6: Supported Base Layer 3 Features

Category, Description, and Configuration Notes	Supported on	
	FESX, FSX, FSX 800, FSX 1600	**FGS, FLS**
802.3ad link aggregation (dynamic trunk groups) • Foundry ports enabled for link aggregation follow the same rules as ports configured for trunk groups. See “Trunk Group Rules” on page 13-3.	X	X
LACP (802.3ad)	X	X
ACL-based rate limiting QoS	X	X
ACLs	X	X
DHCP Relay	X	X
DHCP Client-Based Auto-Configuration		X
DVMRP	X	
IP source guard	X	
DHCP Snooping	X	
ECMP	X	
FDP	X	X
LLDP	X	X
LLDP-MED	X	X
GVRP		X
IGMP v1/v2 Snooping Global	X	X
IGMP v3 Snooping Global	X (*,G)	X (*,G)
IGMP v1/v2/v3 Snooping per VLAN	X	X
IGMP v2/v3 Fast Leave	X	X
IGMP Tracking	X	X
IGMP Filters	X	X
IP helper	X	X
Jumbo frames	X	X
Link Fault Signaling (LFS)	X	X
MAC security - lockdown, limit, and port security	X	X

Table 3.6: Supported Base Layer 3 Features (Continued)

Category, Description, and Configuration Notes	Supported on	
	FESX, FSX, FSX 800, FSX 1600	FGS, FLS
MAC-based VLANS		X
MRP I	X	X
MRP II	X	
PIM Snooping	X	X
Port-based VLANs	X	X
Protocol-based VLANs	X	X
RADIUS, TACACS, AAA	X	X
Outbound rate shaping	X	
Outbound rate limiting	X	X
RIP V1 and V2 • Static RIP support only. The Foundry device with Base Layer 3 does not learn RIP routes from other Layer 3 devices. However, the device does advertise directly connected routes.	X	X
Routing for directly connected IP subnets	X	X
Security login 8021X, MAC-based	X	X
Security login, MAC-based VLANs		X
sFlow	X	X
SNMPv3	X	X
SSH2 Server/Client	X	X
Static IP • Up to 4000 IP route entries for FESX, FSX, FWSX • Up to 1000 hardware entries for FGS, FLS - shared with ACLs and MAC features	X	X
Static Multicast MAC	X	X
STP 802.1d	X	X
STP 802.1s and 802.1w	X	X
STP PVST, PVST+, PVRST+	X	X
Super Aggregated VLAN	X	X
802.1Q-in-Q (tag type 8100 over 8100 encapsulation)	X	X (48-port only)
TCP SYN and ICMP Smurf Protection	X	X
Virtual Cable Tester	X	X

Table 3.6: Supported Base Layer 3 Features (Continued)

Category, Description, and Configuration Notes	Supported on	
	FESX, FSX, FSX 800, FSX 1600	FGS, FLS
Virtual Interfaces • Up to 255 virtual interfaces	X	X
VoIP Autoconfiguration and CDP	X	X
VRRP	X	X
VSRP and VSRP Aware	X	X
Web-based management (Vista)	X	X
Web-based management HTTPS/SSL	X	X

Supported Full Layer 3 Features

Table 3.7 lists the supported Full Layer 3 features for FastIron products.

NOTE: Full Layer 3 features are supported on Premium devices only.

Table 3.7: Supported Full Layer 3 Features

Category, Description, and Configuration Notes	Supported on	
	FESX, FSX, FSX 800, FSX 1600	**FGS, FLS**
6,000 active host routes maximum	X	
BGP4 • BGP4 peers: • The M1, M2, M3, and M4 management modules support 20 BGP4 peers each • The FESX supports 4 BGP4 peers • BGP4 routes: • The M1 management module supports 250K BGP4 routes • The M2, M3, and M4 management modules support 1M BGP4 routes each • The FESX supports 100K BGP4 routes	X	
IGMP V1, V2, and V3	X	
IP	X	
IP multicast (DVMRP, PIM-SM, PIM-DM) • Layer 3 Switches support up to 1024 PIM groups and 1024 DVMRP groups by default	X	
OSPF V2 (IPv4)	X	
Policy-Based Routing (PBR) This feature is not supported on Base Layer 3	X	
RIP V1 and V2	X	
Route-only support • FSX devices support disabling Layer 2 Switching at the CLI Interface level as well as the Global CONFIG level. • FESX supports disabling Layer 2 Switching on an individual interface and on a global basis. • This feature is not supported on virtual interfaces	X	
Route redistribution	X	

Table 3.7: Supported Full Layer 3 Features (Continued)

Category, Description, and Configuration Notes	Supported on	
	FESX, FSX, FSX 800, FSX 1600	FGS, FLS
Routes in hardware maximum: • FSX devices support up to 256,000 routes in hardware • FESX devices support up to 100,000 routes in hardware	X	
VRRP and VRRP-E NOTE: VRRP-E is supported in the full Layer 3 code only. It is not supported in the base Layer 3 code.	X	
VRRP-E slow start timer	X	
VSRP, VRRP, and VRRP-E timer scale	X	

Unsupported Features

Table 3.8 lists the features that are not supported on the FastIron X Series and FGS and FLS devices. If required, these features are available on other Foundry devices.

Table 3.8: Unsupported Features

System Level Features not Supported
• ACL logging of permitted packets. Note that ACL logging is supported for denied packets, which are sent to the CPU for logging.
• Broadcast and multicast MAC filters
• NetFlow
• Outbound ACLs
Layer 2 Features not Supported
• MAC-based VLAN **NOTE:** This feature is not supported on FastIron X Series devices.
• MLD **NOTE:** This feature is not supported on FastIron X Series devices.
• SuperSpan
• VLAN-based priority
Layer 3 Features not Supported
• AppleTalk
• Base Layer 3 features and full Layer 3 features are not supported on the FWSX
• BGP4+
• Foundry Standby Router Protocol (FSRP)
• IPX
• IS-IS
• Multiprotocol Border Gateway Protocol (MBGP)
• Multiprotocol Label Switching (MPLS)
• Multiprotocol Source Discovery Protocol (MSDP)
• Network Address Translation (NAT)

Supported IPv6 Management Features

Table 3.9 shows which IPV6 management features are supported in devices running IPV6 software.

Table 3.9: Supported IPv6 Management Features

Category, Description, and Configuration Notes	Supported on	
	FESX, FSX, FSX 800, FSX 1600, FWSX	FGS, FLS
Link-Local IPv6 Address	X	X
IPv6 Access List	X	X
IPv6 copy	X	X
IPv6 ncopy	X	X
IPv6 debug	X	X
IPv6 ping	X	X
IPv6 traceroute	X	X
DNS server name resolution	X	X
HTTP/HTTPS	X	X
Logging (Syslog)	X	X
RADIUS	X	X
SCP	X	X
SSH	X	X
SNMP v1, v2, v3	X	X
SNTP	X	X
Syslog	X	X
TACACS/TACACS+	X	X
Telnet	X	X
TFTP	X	X
Traps	X	X

Chapter 4
Configuring Basic Software Features

Foundry devices are configured at the factory with default parameters that allow you to begin using the basic features of the system immediately. However, many of the advanced features such as VLANs or routing protocols for the device must first be enabled at the system (global) level before they can be configured. If you use the Command Line Interface (CLI) to configure system parameters, you can find these system level parameters at the Global CONFIG level of the CLI.

NOTE: Before assigning or modifying any router parameters, you must assign the IP subnet (interface) addresses for each port.

NOTE: For information about configuring IP addresses, DNS resolver, DHCP assist, and other IP-related parameters, see the chapter “Configuring IP” on page 29-1.

For information about the Syslog buffer and messages, see the Appendix “Using Syslog” on page A-1.

Configuring Basic System Parameters

The procedures in this section describe how to configure the basic system parameters listed in Table 4.1.

Table 4.1: Basic System Parameters

Basic System Parameter	See Page
System name, contact, and location	4-2
SNMP trap receiver, trap source address, and other parameters	4-2
Single source address for all Telnet packets	4-7
Single source address for all TFTP packets	4-8
Single source address for all Syslog packets	4-8
Single source address for all SNTP packets	4-9
System time using a Simple Network Time Protocol (SNTP) server or local system counter	4-9

Table 4.1: Basic System Parameters (Continued)

Basic System Parameter	See Page
System clock	4-11
Broadcast, multicast, or unknown-unicast limits, if required to support slower third-party devices	4-13
Banners that are displayed on users' terminals when they enter the Privileged EXEC CLI level or access the device through Telnet	4-14

NOTE: For information about the Syslog buffer and messages, see "Using Syslog" on page A-1.

Entering System Administration Information

You can configure a system name, contact, and location for a Foundry device and save the information locally in the configuration file for future reference. This information is not required for system operation but is suggested. When you configure a system name, the name replaces the default system name in the CLI command prompt.

The name, contact, and location each can be up to 32 alphanumeric characters.

Here is an example of how to configure a system name, system contact, and location:

```
FastIron(config)#hostname zappa
zappa(config)#snmp-server contact Support Services
zappa(config)#snmp-server location Centerville
zappa(config)#end
zappa#write memory
```

Syntax: hostname <string>

Syntax: snmp-server contact <string>

Syntax: snmp-server location <string>

The text strings can contain blanks. The SNMP text strings do not require quotation marks when they contain blanks but the host name does.

NOTE: The **chassis name** command does not change the CLI prompt. Instead, the command assigns an administrative ID to the device.

Configuring Simple Network Management Protocol (SNMP) Parameters

Use the procedures in this section to perform the following configuration tasks:

- Specify an SNMP trap receiver.
- Specify a source address and community string for all traps sent by the device.
- Change the holddown time for SNMP traps
- Disable individual SNMP traps. (All traps are enabled by default.)
- Disable traps for CLI access that is authenticated by a local user account, a RADIUS server, or a TACACS/TACACS+ server.

NOTE: To add and modify "get" (read-only) and "set" (read-write) community strings, see the chapter "Securing Access to Management Functions" on page 39-1.

Specifying an SNMP Trap Receiver

You can specify a trap receiver to ensure that all SNMP traps sent by the Foundry device go to the same SNMP trap receiver or set of receivers, typically one or more host devices on the network. When you specify the host, you also specify a community string. The Foundry device sends all the SNMP traps to the specified host(s) and includes the specified community string. Administrators can therefore filter for traps from a Foundry device based on IP address or community string.

When you add a trap receiver, the software automatically encrypts the community string you associate with the receiver when the string is displayed by the CLI or Web management interface. If you want the software to show the community string in the clear, you must explicitly specify this when you add a trap receiver. In either case, the software does not encrypt the string in the SNMP traps sent to the receiver.

To specify the host to which the device sends all SNMP traps, use one of the following methods.

To add a trap receiver and encrypt the display of the community string, enter commands such as the following:

To specify an SNMP trap receiver and change the UDP port that will be used to receive traps, enter a command such as the following:

```
FastIron(config)#snmp-server host 2.2.2.2 0 mypublic port 200
FastIron(config)#write memory
```

Syntax: snmp-server host <ip-addr> [0 | 1] <string> [port <value>]

The <ip-addr> parameter specifies the IP address of the trap receiver.

The **0 | 1** parameter specifies whether you want the software to encrypt the string (**1**) or show the string in the clear (**0**). The default is **0**.

The <string> parameter specifies an SNMP community string configured on the Foundry device. The string can be a read-only string or a read-write string. The string is not used to authenticate access to the trap host but is instead a useful method for filtering traps on the host. For example, if you configure each of your Foundry devices that use the trap host to send a different community string, you can easily distinguish among the traps from different Foundry devices based on the community strings.

The command in the example above adds trap receiver 2.2.2.2 and configures the software to encrypt display of the community string. When you save the new community string to the startup-config file (using the **write memory** command), the software adds the following command to the file:

```
snmp-server host 2.2.2.2 1 <encrypted-string>
```

To add a trap receiver and configure the software to encrypt display of the community string in the CLI and Web management interface, enter commands such as the following:

```
FastIron(config)#snmp-server host 2.2.2.2 0 FastIron-12
FastIron(config)#write memory
```

The **port** <value> parameter allows you to specify which UDP port will be used by the trap receiver. This parameter allows you to configure several trap receivers in a system. With this parameter, IronView Network Manager Network Manager and another network management application can coexist in the same system. Foundry devices can be configured to send copies of traps to more than one network management application.

Specifying a Single Trap Source

You can specify a single trap source to ensure that all SNMP traps sent by the Foundry device use the same source IP address. When you configure the SNMP source address, you specify the Ethernet port, loopback interface, or virtual interface that is the source for the traps. The Foundry device then uses the lowest-numbered IP address configured on the port or interface as the source IP address in the SNMP traps sent by the device.

Identifying a single source IP address for SNMP traps provides the following benefits:

- If your trap receiver is configured to accept traps only from specific links or IP addresses, you can use this feature to simplify configuration of the trap receiver by configuring the Foundry device to always send the traps from the same link or source address.
- If you specify a loopback interface as the single source for SNMP traps, SNMP trap receivers can receive traps regardless of the states of individual links. Thus, if a link to the trap receiver becomes unavailable but

the receiver can be reached through another link, the receiver still receives the trap, and the trap still has the source IP address of the loopback interface.

To specify a port, loopback interface, or virtual interface whose lowest-numbered IP address the Foundry device must use as the source for all SNMP traps sent by the device, use the following CLI method.

To configure the device to send all SNMP traps from the first configured IP address on port 4, enter the following commands:

```
FastIron(config)#snmp trap-source ethernet 4
FastIron(config)#write memory
```

Syntax: snmp-server trap-source loopback <num> | ethernet [<stacknum>]|[<slotnum>]|<portnum> | ve <num>

The <num> parameter is a loopback interface or virtual interface number.

If you specify an Ethernet port, the <slotnum> parameter is required on chassis devices.

If you specify an Ethernet port the [<stacknum>] parameter is required on all FGS and FLS devices.

To specify a loopback interface as the device's SNMP trap source, enter commands such as the following:

```
FastIron(config)#int loopback 1
FastIron(config-lbif-1)#ip address 10.0.0.1/24
FastIron(config-lbif-1)#exit
FastIron(config)#snmp-server trap-source loopback 1
```

The commands in this example configure loopback interface 1, assign IP address 10.00.1/24 to the loopback interface, then designate the interface as the SNMP trap source for this device. Regardless of the port the Foundry device uses to send traps to the receiver, the traps always arrive from the same source IP address.

Setting the SNMP Trap Holddown Time

When a Foundry device starts up, the software waits for Layer 2 convergence (STP) and Layer 3 convergence (OSPF) before beginning to send SNMP traps to external SNMP servers. Until convergence occurs, the device might not be able to reach the servers, in which case the messages are lost.

By default, a Foundry device uses a one-minute holddown time to wait for the convergence to occur before starting to send SNMP traps. After the holddown time expires, the device sends the traps, including traps such as "cold start" or "warm start" that occur before the holddown time expires.

You can change the holddown time to a value from one second to ten minutes.

To change the holddown time for SNMP traps, enter a command such as the following at the global CONFIG level of the CLI:

```
FastIron(config)#snmp-server enable traps holddown-time 30
```

The command in this example changes the holddown time for SNMP traps to 30 seconds. The device waits 30 seconds to allow convergence in STP and OSPF before sending traps to the SNMP trap receiver.

Syntax: [no] snmp-server enable traps holddown-time <secs>

The <secs> parameter specifies the number of seconds and can be from 1 – 600 (ten minutes). The default is 60 seconds.

Disabling SNMP Traps

Foundry devices come with SNMP trap generation enabled by default for all traps. You can selectively disable one or more of the following traps.

NOTE: By default, all SNMP traps are enabled at system startup.

Layer 2 Traps

The following traps are generated on devices running Layer 2 software:

- SNMP authentication keys

- Power supply failure
- Fan failure
- Cold start
- Link up
- Link down
- Bridge new root
- Bridge topology change
- Locked address violation

Layer 3 Traps

The following traps are generated on devices running Layer 3 software:

- SNMP authentication key
- Power supply failure
- Fan failure
- Cold start
- Link up
- Link down
- Bridge new root
- Bridge topology change
- Locked address violation
- BGP4
- OSPF
- VRRP
- VRRPE

To stop link down occurrences from being reported, enter the following:

```
FastIron(config)#no snmp-server enable traps link-down
```

Syntax: [no] snmp-server enable traps <trap-type>

Displaying Virtual Routing Interface Statistics

Platform Support:

- FESX/FSX/FWSX devices running software release 03.2.00 and later – L2, BL3, L3

In software releases prior to 04.1.00, this feature enables SNMP to extract and display virtual routing interface statistics from the ifTable (32-bit counters).

Starting with software release 04.1.00, this feature enables SNMP to extract and display virtual routing interface statistics from the ifXTable (64-bit counters).

The following describes the limitations of this feature:

- The Foundry device counts traffic from all virtual interfaces (VEs). For example, in a configuration with two VLANs (VLAN 1 and VLAN 20) on port 1, when traffic is sent on VLAN 1, the counters (VE statistics) increase for both VE 1 and VE 20.
- The counters include all traffic on each virtual interface, even if the virtual interface is disabled.
- The counters include traffic that is denied by ACLs or MAC filters.

To enable SNMP to display VE statistics, enter the following command:

```
FastIron(config)#enable snmp ve-statistics
```

Syntax: [no] enable snmp ve-statistics

Use the **no** form of the command to disable this feature once it is enabled.

Note that the above CLI command enables SNMP to display virtual interface statistics. It does not enable the CLI or Web Management Interface to display the statistics.

Disabling Syslog Messages and Traps for CLI Access

Foundry devices send Syslog messages and SNMP traps when a user logs into or out of the User EXEC or Privileged EXEC level of the CLI. The feature applies to users whose access is authenticated by an authentication-method list based on a local user account, RADIUS server, or TACACS/TACACS+ server.

NOTE: The Privileged EXEC level is sometimes called the “Enable” level, because the command for accessing this level is **enable**.

The feature is enabled by default.

Examples of Syslog Messages for CLI Access

When a user whose access is authenticated by a local user account, a RADIUS server, or a TACACS/TACACS+ server logs into or out of the CLI’s User EXEC or Privileged EXEC mode, the software generates a Syslog message and trap containing the following information:

- The time stamp
- The user name
- Whether the user logged in or out
- The CLI level the user logged into or out of (User EXEC or Privileged EXEC level)

NOTE: Messages for accessing the User EXEC level apply only to access through Telnet. The device does not authenticate initial access through serial connections but does authenticate serial access to the Privileged EXEC level. Messages for accessing the Privileged EXEC level apply to access through the serial connection or Telnet.

The following examples show login and logout messages for the User EXEC and Privileged EXEC levels of the CLI:

```
FastIron#show logging

Syslog logging: enabled (0 messages dropped, 0 flushes, 0 overruns)
Buffer logging: level ACDMEINW, 12 messages logged
level code: A=alert C=critical D=debugging M=emergency E=error
I=informational N=notification W=warning

Static Log Buffer:
Dec 15 19:04:14:A:Fan 1, fan on right connector, failed

Dynamic Log Buffer (50 entries):
Oct 15 18:01:11:info:dg logout from USER EXEC mode
Oct 15 17:59:22:info:dg logout from PRIVILEGE EXEC mode
Oct 15 17:38:07:info:dg login to PRIVILEGE EXEC mode
Oct 15 17:38:03:info:dg login to USER EXEC mode
```

Syntax: show logging

The first message (the one on the bottom) indicates that user “dg” logged in to the CLI’s User EXEC level on October 15 at 5:38 PM and 3 seconds (Oct 15 17:38:03). The same user logged into the Privileged EXEC level four seconds later.

The user remained in the Privileged EXEC mode until 5:59 PM and 22 seconds. (The user could have used the CONFIG modes as well. Once you access the Privileged EXEC level, no further authentication is required to access the CONFIG levels.) At 6:01 PM and 11 seconds, the user ended the CLI session.

Disabling the Syslog Messages and Traps

Logging of CLI access is enabled by default. If you want to disable the logging, enter the following commands:

```
FastIron(config)#no logging enable user-login
FastIron(config)#write memory
FastIron(config)#end
FastIron#reload
```

Syntax: [no] logging enable user-login

Configuring an Interface as the Source for All Telnet Packets

You can designate the lowest-numbered IP address configured on an interface as the source IP address for all Telnet packets from the device. Identifying a single source IP address for Telnet packets provides the following benefits:

- If your Telnet server is configured to accept packets only from specific links or IP addresses, you can use this feature to simplify configuration of the Telnet server by configuring the Foundry device to always send the Telnet packets from the same link or source address.
- If you specify a loopback interface as the single source for Telnet packets, Telnet servers can receive the packets regardless of the states of individual links. Thus, if a link to the Telnet server becomes unavailable but the client or server can be reached through another link, the client or server still receives the packets, and the packets still have the source IP address of the loopback interface.

The software contains separate CLI commands for specifying the source interface for Telnet, TACACS/ TACACS+, and RADIUS packets. You can configure a source interface for one or more of these types of packets.

To specify an interface as the source for all Telnet packets from the device, use the following CLI method. The software uses the lowest-numbered IP address configured on the interface as the source IP address for Telnet packets originated by the device.

To specify the lowest-numbered IP address configured on a virtual interface as the device’s source for all Telnet packets, enter commands such as the following:

```
FastIron(config)#int loopback 2
FastIron(config-lbif-2)#ip address 10.0.0.2/24
FastIron(config-lbif-2)#exit
FastIron(config)#ip telnet source-interface loopback 2
```

The commands in this example configure loopback interface 2, assign IP address 10.0.0.2/24 to the interface, then designate the interface as the source for all Telnet packets from the device.

Syntax: ip telnet source-interface ethernet [<stacknum>]|[<slotnum>]|<portnum> | loopback <num> | ve <num>

If you specify an Ethernet port, the <slotnum> parameter is required on chassis devices.

If you specify an Ethernet port the[<stacknum> parameter is required on all FGS and FLS devices.

The following commands configure an IP interface on an Ethernet port and designate the address port as the source for all Telnet packets from the device.

```
FastIron(config)#interface ethernet 4
FastIron(config-if-e1000-4)#ip address 209.157.22.110/24
FastIron(config-if-e1000-4)#exit
FastIron(config)#ip telnet source-interface ethernet 4
```

Cancelling an Outbound Telnet Session

If you want to cancel a Telnet session from the console to a remote Telnet server (for example, if the connection is frozen), you can terminate the Telnet session by doing the following:

1. At the console, press Ctrl-^ (Ctrl-Shift-6).
2. Press the X key to terminate the Telnet session.

Pressing Ctrl-^ twice in a row causes a single Ctrl-^ character to be sent to the Telnet server. After you press Ctrl-^, pressing any key other than X or Ctrl-^ returns you to the Telnet session.

Configuring an Interface as the Source for All TFTP Packets

Platform Support:

- FESX/FSX/FWSX devices running software release 02.5.00 and later

You can configure the device to use the lowest-numbered IP or IPv6 address configured on a loopback interface, virtual interface, or Ethernet port as the source for all TFTP packets from the device. The software uses the lowest-numbered IP or IPv6 address configured on the interface as the source IP address for the packets.

For example, to specify the lowest-numbered IP address configured on a virtual interface as the device's source for all TFTP packets, enter commands such as the following:

```
FastIron(config)#int ve 1
FastIron(config-vif-1)#ip address 10.0.0.3/24
FastIron(config-vif-1)#exit

FastIron(config)#ip tftp source-interface ve 1
```

The commands in this example configure virtual interface 1, assign IP address 10.0.0.3/24 to the interface, then designate the interface's address as the source address for all TFTP packets.

Syntax: [no] ip tftp source-interface ethernet [<stacknum>]|[<slotnum>]|<portnum> | loopback <num> | ve <num>

If you specify an Ethernet port, the <slotnum> parameter is required on chassis devices.

If you specify an Ethernet port the <stacknum> parameter is required on all FGS and FLS devices.

The default is the lowest-numbered IP address configured on the port through which the packet is sent. The address therefore changes, by default, depending on the port.

Configuring an Interface as the Source for Syslog Packets

Platform Support:

- FESX/FSX/FWSX devices running software release 02.5.00 and later

You can configure the device to use the lowest-numbered IP or IPv6 address configured on a loopback interface, virtual interface, or Ethernet port as the source for all Syslog packets from the device. The software uses the lowest-numbered IP or IPv6 address configured on the interface as the source IP address for the packets.

For example, to specify the lowest-numbered IP address configured on a virtual interface as the device's source for all Syslog packets, enter commands such as the following:

```
FastIron(config)#int ve 1
FastIron(config-vif-1)#ip address 10.0.0.4/24
FastIron(config-vif-1)#exit
FastIron(config)#ip syslog source-interface ve 1
```

The commands in this example configure virtual interface 1, assign IP address 10.0.0.4/24 to the interface, then designate the interface's address as the source address for all Syslog packets.

Syntax: [no] ip syslog source-interface ethernet [<stacknum>]|[<slotnum>]|<portnum> | loopback <num> | ve <num>

If you specify an Ethernet port, the <slotnum> parameter is required on chassis devices.

If you specify an Ethernet port the <stacknum> parameter is required on all FGS and FLS devices.

The default is the lowest-numbered IP or IPv6 address configured on the port through which the packet is sent. The address therefore changes, by default, depending on the port.

Configuring an Interface as the Source for All SNTP Packets

Platform Support:

- FESX/FSX/FWSX devices running software release 02.5.00 and later

You can configure the device to use the lowest-numbered IP or IPv6 address configured on a loopback interface, virtual interface, or Ethernet port as the source for all SNTP packets from the device. The software uses the lowest-numbered IP or IPv6 address configured on the interface as the source IP address for the packets.

For example, to specify the lowest-numbered IP address configured on a virtual interface as the device's source for all SNTP packets, enter commands such as the following:

```
FastIron(config)#int ve 1
FastIron(config-vif-1)#ip address 10.0.0.5/24
FastIron(config-vif-1)#exit
FastIron(config)#ip sntp source-interface ve 1
```

The commands in this example configure virtual interface 1, assign IP address 10.0.0.5/24 to the interface, then designate the interface's address as the source address for all SNTP packets.

Syntax: [no] ip sntp source-interface ethernet [<slotnum>/]<portnum> | loopback <num> | ve <num>

The default is the lowest-numbered IP or IPv6 address configured on the port through which the packet is sent. The address therefore changes, by default, depending on the port.

Specifying a Simple Network Time Protocol (SNTP) Server

You can configure the Foundry device to consult SNTP servers for the current system time and date.

NOTE: Foundry devices do not retain time and date information across power cycles. Unless you want to reconfigure the system time counter each time the system is reset, Foundry Networks recommends that you use the SNTP feature.

To identify an SNTP server with IP address 208.99.8.95 to act as the clock reference for a Foundry device, enter the following:

```
FastIron(config)#sntp server 208.99.8.95
```

Syntax: sntp server <ip-addr> | <hostname> [<version>]

The <version> parameter specifies the SNTP version the server is running and can be from 1 – 4. The default is 1. You can configure up to three SNTP servers by entering three separate **sntp server** commands.

By default, the Foundry device polls its SNTP server every 30 minutes (1800 seconds). To configure the Foundry device to poll for clock updates from a SNTP server every 15 minutes, enter the following:

```
FastIron(config)#sntp poll-interval 900
```

Syntax: [no] sntp poll-interval <1-65535>

To display information about SNTP associations, enter the following command:

```
FastIron#show sntp associations
  address         ref clock       st  when  poll  delay  disp
 ~207.95.6.102    0.0.0.0         16   202     4    0.0     5.45
 ~207.95.6.101    0.0.0.0         16   202     0    0.0     0.0
* synced, ~ configured
```

Syntax: show sntp associations

The following table describes the information displayed by the **show sntp associations** command.

Table 4.2: Output from the show sntp associations command

This Field...	Displays...
(leading character)	One or both of the following: * Synchronized to this peer ~ Peer is statically configured
address	IP address of the peer
ref clock	IP address of the peer's reference clock
st	NTP stratum level of the peer
when	Amount of time since the last NTP packet was received from the peer
poll	Poll interval in seconds
delay	Round trip delay in milliseconds
disp	Dispersion in seconds

To display information about SNTP status, enter the following command:

```
FastIron#show sntp status
Clock is unsynchronized, stratum = 0, no reference clock
precision is 2**0
reference time is 0       .0
clock offset is 0.0    msec, root delay is 0.0  msec
root dispersion is 0.0  msec, peer dispersion is 0.0  msec
```

Syntax: show sntp status

The following table describes the information displayed by the **show sntp status** command.

Table 4.3: Output from the show sntp status command

This Field...	Indicates...
unsynchronized	System is not synchronized to an NTP peer.
synchronized	System is synchronized to an NTP peer.

Table 4.3: Output from the show sntp status command (Continued)

This Field...	Indicates...
stratum	NTP stratum level of this system
reference clock	IP Address of the peer (if any) to which the unit is synchronized
precision	Precision of this system's clock (in Hz)
reference time	Reference time stamp
clock offset	Offset of clock to synchronized peer
root delay	Total delay along the path to the root clock
root dispersion	Dispersion of the root path
peer dispersion	Dispersion of the synchronized peer

SNTP over IPv6

Platform Support:

- FESX/FSX/FWSX devices running software release 02.4.00 and later
- FGS and FLS devices running software release 04.0.00 and later

To enable the Foundry device to send SNTP packets over IPv6, enter a command such as the following at the Global CONFIG level of the CLI:

```
FastIron(config)#sntp server ipv6 3000::400
```

Syntax: sntp server ipv6 <ipv6-address>

The <ipv6-address> is the IPv6 address of the SNTP server. When you enter the IPv6 address, you do not need to specify the prefix length. A prefix length of 128 is implied.

Setting the System Clock

In addition to SNTP support, Foundry switches and routers also allow you to set the system time counter. The time counter setting is not retained across power cycles and is not automatically synchronized with an SNTP server. The counter merely starts the system time and date clock with the time and date you specify.

NOTE: You can synchronize the time counter with your SNTP server time by entering the **sntp sync** command from the Privileged EXEC level of the CLI.

NOTE: Unless you identify an SNTP server for the system time and date, you will need to re-enter the time and date following each reboot.

For more details about SNTP, see “Specifying a Simple Network Time Protocol (SNTP) Server” on page 4-9.

To set the system time and date to 10:15:05 on October 15, 2003, enter the following command:

```
FastIron#clock set 10:15:05 10-15-2003
```

Syntax: [no] clock set <hh:mm:ss> <mm-dd-yy> | <mm-dd-yyyy>

By default, Foundry switches and routers do not change the system time for daylight saving time. To enable daylight saving time, enter the following command:

```
FastIron#clock summer-time
```

Syntax: clock summer-time

Although SNTP servers typically deliver the time and date in Greenwich Mean Time (GMT), you can configure the Foundry device to adjust the time for any one-hour offset from GMT or for one of the following U.S. time zones:

- US Pacific (default)
- Alaska
- Aleutian
- Arizona
- Central
- East-Indiana
- Eastern
- Hawaii
- Michigan
- Mountain
- Pacific
- Samoa

The default is US Pacific.

To change the time zone to Australian East Coast time (which is normally 10 hours ahead of GMT), enter the following command:

```
FastIron(config)#clock timezone gmt+10
```

Syntax: clock timezone gmt | us <time-zone>

You can enter one of the following values for <time-zone>:

- US time zones (**us**): alaska, aleutian, arizona, central, east-indiana, eastern, hawaii, michigan, mountain, pacific, samoa.
- GMT time zones (gmt): gmt+0:00 to gmt+12:00 in increments of 1, and gmt-0:00 to gmt-12:00 in decrements of 1 are supported.
- FGS Release 03.0.00 adds support for the following additional time zones: gmt+11:30, gmt+10:30, gmt+09:30, gmt+06:30, gmt+05:30, gmt+04:30, gmt+03:30, gmt-03:30, gmt-08:30, gmt-09:30.

New Start and End Dates for US Daylight Saving Time

NOTE: This feature applies to US time zones only.

Starting in 2007, Foundry's software will automatically change the system clock to Daylight Saving Time (DST), in compliance with the new federally mandated start of daylight saving time, which is extended one month beginning in 2007. The DST will start at 2:00am on the second Sunday in March and will end at 2:00am on the first Sunday in November.

The DST feature is automatic, but to trigger the device to the correct time, the device must be configured to the US time zone, not the GMT offset. To configure your device to use the US time zone, enter the following command:

```
FastIron(config)#clock timezone us pacific
```

Syntax: [no] clock timezone us <timezone-type>

Enter pacific, eastern, central, or mountain for <timezone-type>.

This command must be configured on every device that follows the US DST.

To verify the change, run a **show clock** command:

```
FastIron#show clock
```

Refer to October 19, 2006 - Daylight Saving Time 2007 Advisory, posted on kp.foundrynet.com for more information

Limiting Broadcast, Multicast, and Unknown Unicast Traffic

FastIron devices can forward all flooded traffic at wire speed within a VLAN. However, some third-party networking devices cannot handle high rates of broadcast, multicast, or unknown-unicast traffic. If high rates of traffic are being received by the FastIron on a given port of that VLAN, you can limit the number of broadcast, multicast, or unknown-unicast packets or bytes received each second on that port. This can help to control the number of such packets or bytes that are flooded on the VLAN to other devices.

Byte-based limiting for broadcast, multicast, and unknown unicast traffic provides the ability to rate limit traffic based on byte count instead of packet count. When the byte mode is enabled, packets will be received on a port as long as the number of bytes received per second is less than the corresponding limit. Once the limit is reached, further packets will be dropped.

Packet-based and byte-based limiting can be configured simultaneously on the same port. For example, you can configure the broadcast limit in packet mode and the unknown unicast limit in the byte mode on the same port.

When you enable broadcast limiting, the total number of broadcast packets or bytes received on the port will not exceed the number you specify. To also limit multicast packets, enable them after you enable broadcast limiting. In this case, the total number of broadcast and multicast packets or bytes received on the port will not exceed the number you specify.

On FastIron devices, unknown unicast limiting is independent of broadcast and multicast limiting.

Configuration Considerations for FastIron X Series Devices

On FastIron X Series devices, when you configure unknown-unicast limiting, the rate applies to all ports in the ***port range*** for which unknown unicast is enabled. A 1-Gigabit port range consists of 12 ports. For example, the FESX424 has 2 port ranges; ports 1 – 12 are one port range, and ports 13 – 24 are another port range. If you enable unknown unicast limiting on port 2, the configuration applies to the ports from 1 – 12 that have unknown unicast limiting enabled. 10-Gigabit ports are not grouped into ranges. So if your device has two 10-Gigabit uplinks, you can configure different unknown-unicast limits for each 10-Gigabit port.

Command Syntax for Packet-based Limiting

Platform Support:

- FGS and FLS – all software releases
- FESX/FSX/FWSX – all software releases

To enable broadcast limiting on a group of ports by counting the number of packets received, enter commands such as the following:

```
FastIron(config)#interface ethernet 1 to 8
FastIron(config-mif-e1000-1-8)#broadcast limit 65536
```

These commands configure packet-based broadcast limiting on ports 1 – 8. On each port, the total combined number of broadcast packets per second cannot exceed 65,536.

To include multicasts in the 65536 packets per second limit on each of the ports, enter the following command after enabling broadcast limiting:

```
FastIron(config-mif-e1000-1-8)#multicast limit
```

To enable unknown unicast limiting by counting the number of packets received, enter commands such as the following:

```
FastIron#config terminal
FastIron(config)#int e 1
FastIron(config-if-e1000-1)#unknown unicast limit 65536
The combined number of inbound Unknown Unicast packets permitted
    for ports 1 to 12 is now set to 65536
```

```
FastIron((config-if-e1000-1)#
```

Syntax: [no] broadcast limit <num>

Syntax: [no] multicast limit

Syntax: [no]unknown unicast limit <num>

The <num> parameter specifies the maximum number of packets per second and can be any number that is a multiple of 65536, up to a maximum value of 2147418112. If you enter the **multicast limit** command, multicast packets are included in the limit you specify. If you specify 0, limiting is disabled. If you specify a number that is not a multiple of 65536, the software rounds the number to the next multiple of 65536. Limiting is disabled by default.

Command Syntax for Byte-based Limiting

Platform Support:

- FESX/FSX/FWSX devices running software release 04.0.00 and later – L2, BL3, L3

To enable broadcast limiting on a group of ports by counting the number of bytes received, enter commands such as the following:

```
FastIron(config)#interface ethernet 9 to 10
FastIron(config-mif-e1000-9-10)#broadcast limit 131072 bytes
```

These commands configure byte-based broadcast limiting on ports 9 and 10. On each port, the total number of bytes received from broadcast packets cannot exceed 131,072 per second.

To include multicasts in the 131072 bytes per second limit on each of the ports, enter the following command after enabling broadcast limiting:

```
FastIron(config-mif-e1000-1-8)#multicast limit
```

To enable unknown unicast limiting, enter commands such as the following:

```
FastIron#config terminal
FastIron(config)#int e 13
FastIron(config-if-e1000-13)#unknown unicast limit 65536 bytes
The combined number of bytes of inbound Unknown Unicast packets
     permitted for ports 13 to 24 is now set to 65536
FastIron((config-if-e1000-13)#
```

Syntax: [no] broadcast limit <num> bytes

Syntax: [no] multicast limit

Syntax: [no]unknown unicast limit <num> bytes

The <num> parameter specifies the maximum number of bytes per second and can be any number that is a multiple of 65536, up to a maximum value of 2147418112. If you enter the **multicast limit** command, multicast packets are included in the limit you specify. If you specify 0, limiting is disabled. If you specify a number that is not a multiple of 65536, the software rounds the number to the next multiple of 65536. Limiting is disabled by default.

Configuring CLI Banners

Foundry devices can be configured to display a greeting message on users' terminals when they enter the Privileged EXEC CLI level or access the device through Telnet. In addition, a Foundry device can display a message on the Console when an incoming Telnet CLI session is detected.

Setting a Message of the Day Banner

You can configure the Foundry device to display a message on a user's terminal when he or she establishes a Telnet CLI session. For example, to display the message "Welcome to FESX!" when a Telnet CLI session is established:

```
FastIron(config)#banner motd $ (Press Return)
```

```
Enter TEXT message, End with the character '$'.
Welcome to FESX! $
```

A delimiting character is established on the first line of the **banner motd** command. You begin and end the message with this delimiting character. The delimiting character can be any character except " (double-quotation mark) and cannot appear in the banner text. In this example, the delimiting character is $ (dollar sign). The text in between the dollar signs is the contents of the banner. The banner text can be up to 2048 characters long and can consist of multiple lines. To remove the banner, enter the **no banner motd** command.

Syntax: [no] banner <delimiting-character> | [motd <delimiting-character>]

NOTE: The **banner** <delimiting-character> command is equivalent to the **banner motd** <delimiting-character> command.

When you access the Web management interface, the banner is displayed:

Requiring Users to Press the Enter Key after the Message of the Day Banner

In releases prior to 03.0.01a for FastIron X Series devices, users were required to press the Enter key after the Message of the Day (MOTD) was displayed, prior to logging in to the Foundry device on a console or via a Telnet session. Beginning with release 03.0.01a, this requirement is disabled by default. Unless configured, users do not have to press Enter after the MOTD banner is displayed.

For example, if the MOTD "Authorized Access Only" is configured, by default, the following messages are displayed when a user access the Foundry device via Telnet:

```
Authorized Access Only ...

Username:
```

The user can then login to the device.

However, if the requirement to press the Enter key is enabled, the following messages are displayed when accessing the switch via Telnet:

```
Authorized Access Only ...

Press <Enter> to accept and continue the login process....
```

The user must press the Enter key before the login prompt is displayed.

Also, on the console, the following messages are displayed if the requirement to press the Enter key is disabled:

```
Press Enter key to login

Authorized Access Only ...

User Access Verification

Please Enter Login Name:
```

However, if the requirement to press the Enter key after a MOTD is enabled, the following messages are displayed when accessing the switch on the console:

```
Press Enter key to login

Authorized Access Only ...

Press <Enter> to accept and continue the login process....
```

The user must press the Enter key to continue to the login prompt.

To enable the requirement to press the Enter key after the MOTD is displayed, enter a command such as the following:

```
FastIron(config)#banner motd require-enter-key
```

Syntax: [no] banner motd require-enter-key

Use the **no** form of the command to disable the requirement.

Setting a Privileged EXEC CLI Level Banner

You can configure the Foundry device to display a message when a user enters the Privileged EXEC CLI level. For example:

```
FastIron(config)#banner exec_mode #(Press Return)
Enter TEXT message, End with the character '#'.
You are entering Privileged EXEC level
Don’t foul anything up! #
```

As with the **banner motd** command, you begin and end the message with a delimiting character; in this example, the delimiting character is #(pound sign). To remove the banner, enter the **no banner exec_mode** command.

Syntax: [no] banner exec_mode <delimiting-character>

Displaying a Console Message when an Incoming Telnet Session Is Detected

You can configure the Foundry device to display a message on the Console when a user establishes a Telnet session. This message indicates where the user is connecting from and displays a configurable text message.

For example:

```
FastIron(config)#banner incoming $ (Press Return)
Enter TEXT message, End with the character '$'.
Incoming Telnet Session!! $
```

When a user connects to the CLI using Telnet, the following message appears on the Console:

```
Telnet from 209.157.22.63
Incoming Telnet Session!!
```

Syntax: [no] banner incoming <delimiting-character>

To remove the banner, enter the **no banner incoming** command.

Configuring Basic Port Parameters

The procedures in this section describe how to configure the port parameters shown in Table 4.4

Table 4.4: Basic Port Parameters

Port Parameter	See Page
Name	4-17
Speed	4-17
Auto-negotiation maximum port speed advertisement and down-shift	4-18
Duplex mode	4-20
MDI/MDIX detection	4-21
Port status (enable or disable)	4-22
Flow control	4-22

Table 4.4: Basic Port Parameters (Continued)

Port Parameter	See Page
Auto-negotiation and advertisement of flow control	4-22
Configuring PHY FIFO Rx and TX Depth	4-24
Interpacket Gap (IPG)	4-25
Gigabit fiber negotiate mode	4-28
QoS priority	4-28
Dynamic configuration of Voice over IP (VoIP) phones	4-29
Port flap dampening	4-30

All Foundry ports are pre-configured with default values that allow the device to be fully operational at initial startup without any additional configuration. However, in some cases, changes to the port parameters may be necessary to adjust to attached devices or other network requirements.

Assigning a Port Name

A port name can be assigned to help identify interfaces on the network. You can assign a port name to physical ports, virtual interfaces, and loopback interfaces.

To assign a name to a port:

```
FastIron(config)#interface e 2
FastIron(config-if-e1000-2)#port-name Marsha
```

Syntax: port-name <text>

The <text> parameter is an alphanumeric string. The name can be up to 64 characters long. The name can contain blanks. You do not need to use quotation marks around the string, even when it contains blanks.

Modifying Port Speed and Duplex Mode

The Gigabit Ethernet copper ports on the Foundry device are designed to auto-sense and auto-negotiate the speed and duplex mode of the connected device. If the attached device does not support this operation, you can manually enter the port speed to operate at either 10, 100, or 1000 Mbps. The default and recommended setting is 10/100/1000 auto-sense.

NOTE: You can modify the port speed of copper ports only. This feature does not apply to fiber ports. For optimal link operation, link ports on devices that do not support 803.3u must be configured with like parameters, such as speed (10,100,1000), duplex (half, full), MDI/MDIX, and Flow Control.

Configuration Syntax

The following commands change the port speed of interface 8 from the default of 10/100/1000 auto-sense, to 100 Mbps operating in full-duplex mode.

```
FastIron(config)#interface e 8
FastIron(config-if-e1000-8)#speed-duplex 100-full
```

Syntax: speed-duplex <value>

where <value> can be one of the following:

- 10-full
- 10-half
- 100-full

- 100-half
- 1000-full-master
- 1000-full-slave
- auto

The default is auto (auto-negotiation).

Use the **no** form of the command to restore the default.

NOTE: When setting the speed and duplex-mode of an interface to 1000-full, configure one side of the link as master (**1000-full-master**) and the other side as slave (**1000-full-slave**).

NOTE: For optimal link operation, link ports on devices that do not support 803.3u must be configured with like parameters, such as speed (10,100,1000), duplex (half, full), MDI/MDIX, and Flow Control.

Enabling Auto-Negotiation Maximum Port Speed Advertisement and Down-Shift

Platform Support:

- FESX/FSX/FWSX devices running software release 2.3.01 and later
- FGS and FLS devices running software release 02.5.00 and later

NOTE: For optimal link operation, link ports on devices that do not support 803.3u must be configured with like parameters, such as speed (10,100,1000), duplex (half, full), MDI/MDIX, and Flow Control.

Maximum Port speed advertisement and ***Port speed down-shift*** are enhancements to the auto-negotiation feature, a mechanism for accommodating multi-speed network devices by automatically configuring the highest performance mode of inter-operation between two connected devices.

- ***Port speed down-shift*** enables Gigabit copper ports on the Foundry device to establish a link at 1000 Mbps over a 4-pair wire when possible, or to down-shift to 100 Mbps if the medium is a 2-pair wire.
- ***Maximum port speed advertisement*** enables you to configure an auto-negotiation maximum speed that Gigabit copper ports on the Foundry device will advertise to the connected device. You can configure a port to advertise a maximum speed of either 100 Mbps or 10 Mbps. When the maximum port speed advertisement feature is configured on a port that is operating at 100 Mbps maximum speed, the port will advertise 10/100 Mbps capability to the connected device. Similarly, if a port is configured at 10 Mbps maximum speed, the port will advertise 10 Mbps capability to the connected device.

The port speed down-shift and maximum port speed advertisement features operate dynamically at the physical link layer between two connected network devices. They examine the cabling conditions and the physical capabilities of the remote link, then configure the speed of the link segment according to the highest physical-layer technology that both devices can accommodate.

The port speed down-shift and maximum port speed advertisement features operate dynamically at the physical link layer, independent of logical trunk group configurations. Although Foundry recommends that you use the same cable types and auto-negotiation configuration on all members of a trunk group, you could utilize the auto-negotiation features conducive to your cabling environment. For example, in certain circumstances, you could configure each port in a trunk group to have its own auto-negotiation maximum port speed advertisement or port speed down-shift configuration.

Application Notes

- Port speed down-shift and maximum port speed advertisement work only when auto-negotiation is enabled (CLI command **speed-duplex auto**). If auto-negotiation is OFF, the device will reject the port speed down-shift and maximum port speed advertisement configuration.

- When port speed down-shift or maximum port speed advertisement is enabled on a port, the device will reject any configuration attempts to set the port to a forced speed mode (100 Mbps or 1000 Mbps).
- When the port speed down-shift feature is enabled on a combo port, the port will not support true media automatic detection, meaning the device will not be able to detect and select the fiber or copper connector based on link availability.

Enabling Port Speed Down-Shift

Platform Support:

- FESX/FSX/FWSX devices running software release 04.0.00 and later
- FGS and FLS devices running software release 04.0.00 and later

To enable port speed down-shift on a port that has auto-negotiation enabled, enter a command such as the following at the Global CONFIG level of the CLI:

```
FastIron(config)#link-config gig copper autoneg-control down-shift e 1 e 2
```

The above command configures Gigabit copper ports 1 and 2 to establish a link at 1000 Mbps over a 4-pair wire when possible, or to down-shift (reduce the speed) to 100 Mbps when the medium is a 2-pair wire.

Syntax: [no] link-config gig copper autoneg-control down-shift ethernet [<stacknum>/<slotnum>/]<portnum>] [ethernet [<stacknum>/<slotnum>/]<portnum>]

The <stacknum> parameter is required on FastIron GS devices running software release 02.5.00 or later.

The <slotnum> parameter is required on chassis devices and on FastIron GS devices running software release 02.5.00 or later.

The <portnum> parameter is a valid port number.

You can enable port speed down-shift on one or two ports at a time.

To disable port speed down-shift after it has been enabled, enter the **no** form of the command.

Configuring Port Speed Down-Shift and Auto-Negotiation for a Range of Ports

Port speed down-shift and auto-negotiation can be configured for an entire range of ports with a single command.

For example, to configure down-shift on ports 0/1/1 to 0/1/10 and 0/1/15 to 0/1/20 on the FGS, enter:

```
FastIron(config)#link-config gig copper autoneg-control down-shift ethe 0/1/1 to 0/
1/10 ethe 0/1/15 to 0/1/20
```

To configure down-shift on ports 5 to 13 and 17 to 19 on the FESX, enter:

```
FastIron(config)#link-config gig copper autoneg-control down-shift ethe 5 to 13 ethe
17 to 19
```

Syntax: [no] link-config gig copper autoneg-control [down-shift | 100m-auto | 10m-auto> <port-list>

The <port-list> is the list of ports to which the command will be applied.

For <port-list>, specify the port(s) in one of the following formats:

- FastIron GS and LS compact switches – <stacknum/slotnum/portnum>
- FastIron chassis devices – <slotnum/portnum>
- FESX, and FWSX compact switches – <portnum>

You can list all of the ports individually, use the keyword **to** to specify ranges of ports, or a combination of both. To apply the configuration to all ports on the device, use the keyword **all** instead of listing the ports individually.

The output from the **show run** command for this configuration will resemble the following:

```
FastIron#show run
Current configuration:
!
ver 04.0.00b64T7el
!
module 1 fgs-48-port-management-module
module 2 fgs-cx4-2-port-10g-module
!
link-config gig copper autoneg-control down-shift ethe 0/1/1 to 0/1/10 ethe 0/1/15
to 0/1/20
!
!
ip address 10.44.9.11 255.255.255.0
ip default-gateway 10.44.9.1
!
end
```

To disable selective auto-negotiation of 100m-auto on ports 0/1/21 to 0/1/25 and 0/1/30, enter:

```
FastIron(config)#no link-config gig copper autoneg-control 100m-auto ethe 0/1/21 to
0/1/25 ethe 0/1/30
```

NOTE: This feature works with Layer 2 and Layer 3 images.

Configuring Maximum Port Speed Advertisement

To configure a maximum port speed advertisement of 10 Mbps on a port that has auto-negotiation enabled, enter a command such as the following at the Global CONFIG level of the CLI:

```
FastIron(config)#link-config gig copper autoneg-control 10m e 1
```

To configure a maximum port speed advertisement of 100 Mbps on a port that has auto-negotiation enabled, enter the following command at the Global CONFIG level of the CLI:

```
FastIron(config)#link-config gig copper autoneg-control 100m e 2
```

Syntax: [no] link-config gig copper autoneg-control 10m | 100m ethernet [<stacknum>/<slotnum>/]<portnum> [ethernet [<stacknum>/<slotnum>/]<portnum>]

The <stacknum> parameter is required on FastIron GS devices running software release 02.5.00 or later.

The <slotnum> parameter is required on chassis devices and on FastIron GS devices running software release 02.5.00 or later.

The <portnum> parameter is a valid port number.

You can enable maximum port speed advertisement on one or two ports at a time.

To disable maximum port speed advertisement after it has been enabled, enter the **no** form of the command.

Modifying Port Duplex Mode

You can manually configure a 10/100 Mbps port to accept either full-duplex (bi-directional) or half-duplex (uni-directional) traffic.

NOTE: You can modify the port duplex mode of copper ports only. This feature does not apply to fiber ports.

Port duplex mode and port speed are modified by the same command.

Configuration Syntax

To change the port speed of interface 8 from the default of 10/100/1000 auto-sense to 10 Mbps operating at full-duplex, enter the following:

```
FastIron(config)#interface e 8
FastIron(config-if-e1000-8)#speed-duplex 10-full
```

Syntax: speed-duplex <value>

The <value> can be one of the following:

- 10-full
- 10-half
- 100-full
- 100-half
- auto (default)

Configuring MDI/MDIX

The Foundry FastIron devices support automatic Media Dependent Interface (MDI) and Media Dependent Interface Crossover (MDIX) detection on all Gigabit Ethernet Copper ports.

MDI/MDIX is a type of Ethernet port connection using twisted pair cabling. The standard wiring for end stations is MDI, whereas the standard wiring for hubs and switches is MDIX. MDI ports connect to MDIX ports using straight-through twisted pair cabling. For example, an end station connected to a hub or a switch uses a straight-through cable. MDI-to-MDI and MDIX-to-MDIX connections use crossover twisted pair cabling. So, two end stations connected to each other, or two hubs or switches connected to each other, use crossover cable.

The auto MDI/MDIX detection feature can automatically correct errors in cable selection, making the distinction between a straight-through cable and a crossover cable insignificant.

Configuration Notes

- This feature applies to copper ports only.
- The **mdi-mdix auto** command works only when auto-negotiation is ON. If auto-negotiation is OFF and you enter the command **mdi-mdix auto**, the device automatically resets the port to an MDIX only port. In this case, although the Foundry device does not apply the **mdi-mdix auto** configuration, it accepts and saves it. Consequently, when auto-negotiation is turned back ON, the Foundry device applies the **mdi-mdix auto** configuration.
- The **mdi-mdix mdi** and **mdi-mdix mdix** commands work independently of auto-negotiation. Thus, these commands work whether auto-negotiation is turned ON or OFF.
- Do not use the **mdi-mdix** commands on ports that are manually configured with a speed/duplex of **100-full**. In this case, make sure the other port (remote end of the connection) is also configured to 100-full and a cross-over cable is used if the connected device is another switch, hub, or router, or a straight-through cable if the connected device is a host NIC.

Configuration Syntax

The auto MDI/MDIX detection feature is enabled on all Gigabit copper ports by default. For each port, you can disable auto MDI/MDIX, designate the port as an MDI port, or designate the port as an MDIX port.

`To turn off automatic MDI/MDIX detection and define a port as an MDI only port:`

```
FastIron(config-if-e1000-2)#mdi-mdix mdi
```

`To turn off automatic MDI/MDIX detection and define a port as an MDIX only port:`

```
FastIron(config-if-e1000-2)#mdi-mdix mdix
```

`To turn on automatic MDI/MDIX detection on a port that was previously set as an MDI or MDIX port:`

```
FastIron(config-if-e1000-2)#mdi-mdix auto
```

Syntax: mdi-mdix <mdi | mdix | auto>

After you enter the **mdi-mdix** command, the Foundry device resets the port and applies the change.

To display the MDI/MDIX settings, including the configured value and the actual resolved setting (for **mdi-mdix auto**), enter the command **show interface** at any level of the CLI.

Disabling or Re-Enabling a Port

A port can be made inactive (disable) or active (enable) by selecting the appropriate status option. The default value for a port is enabled.

To disable port 8 of a Foundry device, enter the following:

```
FastIron(config)#interface e 8
FastIron(config-if-e1000-8)#disable
```

Syntax: disable

You also can disable or re-enable a virtual interface. To do so, enter commands such as the following:

```
FastIron(config)#interface ve v1
FastIron(config-vif-1)#disable
```

Syntax: disable

To re-enable a virtual interface, enter the **enable** command at the Interface configuration level. For example, to re-enable virtual interface v1, enter the following command:

```
FastIron(config-vif-1)#enable
```

Syntax: enable

Disabling or Re-Enabling Flow Control

You can configure full-duplex ports on a system to operate with or without flow control (802.3x). Flow control is enabled by default.

To disable flow control on full-duplex ports on a system, enter the following:

```
FastIron(config)#no flow-control
```

To turn the feature back on:

```
FastIron(config)#flow-control
```

Syntax: [no] flow-control

NOTE: For optimal link operation, link ports on devices that do not support 803.3u must be configured with like parameters, such as speed (10,100,1000), duplex (half, full), MDI/MDIX, and Flow Control.

Auto-Negotiation and Advertisement of Flow Control

Platform Support:

- FESX/FSX/FWSX devices running software release 03.2.00 and later
- FGS and FLS devices running software release 03.0.00 and later

Auto-negotiation of flow control can be enabled and advertised for 10/100/1000M ports. To enable and advertise flow control capability, enter the following commands:

```
FastIron(config)#interface ethernet 0/1/21

FastIron(config-if-e1000-0/1/21)#flow-control
```

To also enable auto-negotiation of flow control, enter the following commands:

```
FastIron(config)#interface ethernet 0/1/21
```

```
FastIron(config-if-e1000-0/1/21)#flow-control neg-on
```

Syntax: #[no] flow-control [neg-on]

- **flow-control** [default] - Enable flow control, advertise flow control and disable negotiation of flow control
- **flow-control neg-on** - Advertise flow control and enable negotiation of flow control
- **no flow-control** - Disable flow control, disable advertising flow control and also disable negotiation of flow control

Commands may be entered in IF (single port) or MIF (multiple ports at once) mode. For example, enter:

```
FastIron(config)#interface ethernet 0/1/21
```

```
FastIron(config-if-e1000-0/1/21)#flow-control
```

This command enables flow-control on port 0/1/21.

```
FastIron(config)#interface e 0/1/11 to 0/1/15
```

```
FastIron(config-mif-0/1/11-0/1/15)#flow-control
```

This command enables flow-control on ports 0/1/11 to 0/1/15.

Displaying Flow-Control Status

The **show interface** <port> command displays configuration, operation, and negotiation status where applicable.

For example, on a device running the FGS software release 03.0.00 and later, issuing the command for 10/100/1000M port 0/1/21 displays the following output:

```
FastIron#show interfaces ethernet 0/1/21
```

```
GigabitEthernet0/1/21 is up, line protocol is up
  Hardware is GigabitEthernet, address is 00e0.5204.4014 (bia 00e0.5204.4014)
  Configured speed auto, actual 100Mbit, configured duplex fdx, actual fdx
  Configured mdi mode AUTO, actual MDIX
  Member of L2 VLAN ID 1, port is untagged, port state is LISTENING
  BPDU Guard is disabled, Root Protect is disabled
  STP configured to ON, priority is level0
  Flow Control is config enabled, oper enabled, negotiation disabled
  Mirror disabled, Monitor disabled
  Not member of any active trunks
  Not member of any configured trunks
  No port name
  Inter-Packet Gap (IPG) is 96 bit times
  300 second input rate: 0 bits/sec, 0 packets/sec, 0.00% utilization
  300 second output rate: 0 bits/sec, 0 packets/sec, 0.00% utilization
  0 packets input, 0 bytes, 0 no buffer
  Received 0 broadcasts, 0 multicasts, 0 unicasts
  0 input errors, 0 CRC, 0 frame, 0 ignored
  0 runts, 0 giants
  5 packets output, 320 bytes, 0 underruns
  Transmitted 0 broadcasts, 5 multicasts, 0 unicasts
  0 output errors, 0 collisions
```

Issuing the command on a device running the FSX software release 03.2.00 and later displays the following output:

```
FastIron#show interface ethernet 18/1
GigabitEthernet18/1 is up, line protocol is up
Hardware is GigabitEthernet, address is 0012.f228.0600 (bia 0012.f228.0798)
Configured speed auto, actual 1Gbit, configured duplex fdx, actual fdx
Configured mdi mode AUTO, actual MDIX
```

```
Member of 4 L2 VLANs, port is tagged, port state is FORWARDING
BPDU guard is Disabled, ROOT protect is Disabled
Link Error Dampening is Disabled
STP configured to ON, priority is level0, flow control enabled
Flow Control is config enabled, oper enabled, negotiation disabled
mirror disabled, monitor disabled
Not member of any active trunks
Not member of any configured trunks
No port name
IPG MII 96 bits-time, IPG GMII 96 bits-time
IP MTU 1500 bytes, encapsulation ethernet
300 second input rate: 0 bits/sec, 0 packets/sec, 0.00% utilization
300 second output rate: 848 bits/sec, 0 packets/sec, 0.00% utilization
0 packets input, 0 bytes, 0 no buffer
Received 0 broadcasts, 0 multicasts, 0 unicasts
0 input errors, 0 CRC, 0 frame, 0 ignored
0 runts, 0 giants
10251 packets output, 1526444 bytes, 0 underruns
Transmitted 1929 broadcasts, 8293 multicasts, 29 unicasts
0 output errors, 0 collisions
```

The line highlighted in bold will resemble one of the following, depending on the configuration:

- If flow control and auto-negotiation are enabled (and a neighbor does not negotiate flow control), the display shows:

  ```
  Flow Control is config enabled, oper disabled, negotiation enabled
  ```

- If flow control is enabled, and auto-negotiation is disabled, the output shows:

  ```
  Flow Control is config enabled, oper enabled, negotiation disabled
  ```

- If flow control is disabled, the display shows:

  ```
  Flow Control is config disabled, oper disabled
  ```

NOTE: For 10 Gigabit ports, the display shows Flow Control is enabled, or Flow Control is disabled, depending on the configuration.

NOTE: Auto-negotiation of flow control is not supported on 10 Gigabit ports and copper/fiber combination ports.

NOTE: When any of the commands are applied to a port that is up, the port will be disabled and re-enabled.

NOTE: When flow-control is enabled, the hardware can only advertise Pause. It does not advertise Asym.

Configuring PHY FIFO Rx and Tx Depth

Platform Support:

- FGS and FLS devices running software release 03.1.00 and later

FGS Release 03.1.00 adds a new command to increase the FIFO (First In, First Out) depth to adjust for clock differences between connecting devices, if necessary.

PHY devices on FGS and FLS models contain transmit and receive synchronizing FIFOs to adjust for frequency differences between clocks. The **phy-fifo-depth** command allows you to configure the depth of the transmit and receive FIFOs. There are 4 settings (0-3) with 0 as the default. A higher setting indicates a deeper FIFO.

The default setting works for most connections. However, if the clock differences are greater than the default will handle, CRCs and errors will begin to appear on the ports. Raising the FIFO depth setting will adjust for clock differences.

Foundry recommends that you disable the port before applying this command, and re-enable the port. Applying the command while traffic is flowing through the port can cause CRC and other errors for any packets that are actually passing through the PHY while the command is being applied.

Syntax: [no] phy-fifo-depth <setting>

- <setting> is a value between 0 and 3. (0 is the default.)

This command can be issued for a single port from the IF config mode or for multiple ports from the MIF config mode.

NOTE: Higher settings give better tolerance for clock differences with the partner phy, but may marginally increase latency as well.

Configuring the Interpacket Gap (IPG)

Platform Support:

- FESX/FSX/FWSX devices running software release 03.0.00 and later
- FGS and FLS devices running software release 03.0.00 and later (see “Configuring IPG on a FastIron GS and FastIron LS” on page 4-26)

IPG is the time delay, in bit time, between frames transmitted by the device. You configure IPG at the interface level. The command you use depends on the interface type on which IPG is being configured.

The default interpacket gap is 96 bits-time, which is 9.6 microseconds for 10 Mbps Ethernet, 960 nanoseconds for 100 Mbps Ethernet, 96 nanoseconds for 1 Gbps Ethernet, and 9.6 nanoseconds for 10 Gbps Ethernet.

Configuration Notes

When configuring IPG, note the following:

- IPG configuration commands are based on "port regions". All ports within the same port region should have the same IPG configuration. If a port region contains two or more ports, changes to the IPG configuration for one port are applied to all ports in the same port region. When you enter a value for IPG, the CLI displays the ports to which the IPG configuration is applied. For example:

  ```
  FastIron(config-if-e1000-7/1)#ipg-gmii 120
  IPG 120(112) has been successfully configured for ports 7/1 to 7/12
  ```

- When you enter a value for IPG, the device applies the closest valid IPG value for the port mode to the interface. For example, if you specify 120 for a 1 Gigabit Ethernet port in 1 Gigabit mode, the device assigns 112 as the closest valid IPG value to program into hardware.

Configuring IPG on a Gigabit Ethernet Port

On a Gigabit Ethernet port, you can configure IPG for 10/100 mode and for Gigabit Ethernet mode.

10/100M mode

To configure IPG on a Gigabit Ethernet port for 10/100M mode, enter the following command.

```
FastIron(config)#interface ethernet 7/1
FastIron(config-if-e1000-7/1)#ipg-mii 120
IPG 120(120) has been successfully configured for ports 7/1 to 7/12
```

Syntax: [no] ipg-mii <bit time>

Enter 12-124 for <bit time>. The default is 96 bit time.

1G Mode

To configure IPG on a Gigabit Ethernet port for 1-Gigabit Ethernet mode, enter commands such as the following:

```
FastIron(config)#interface ethernet 7/1
FastIron(config-if-e1000-7/1)#ipg-gmii 120
IPG 120(112) has been successfully configured for ports 7/1 to 7/12
```

Syntax: [no] ipg-gmii <bit time>

Enter 48 - 112 for <bit time>. The default is 96 bit time.

Configuring IPG on a 10-Gigabit Ethernet Interface

To configure IPG on a 10-Gigabit Ethernet interface, enter commands such as the following:

```
FastIron(config)#interface ethernet 9/1
FastIron(config-if-e10000-9/1)#ipg-xgmii 120
IPG 120(128) has been successfully configured for port 9/1
```

Syntax: [no] ipg-xgmii <bit time>

Enter 96-192 for <bit time>. The default is 96 bit time.

Configuring IPG on a FastIron GS and FastIron LS

Platform Support:

- FGS and FLS devices running software release 03.0.00

On FGS and FLS devices, you can configure an IPG for each port. An IPG is a configurable time delay between successive data packets.

You can configure an IPG with a range from 48-120 bit times in multiples of 8, with a default of 96. The IPG may be set from either the interface configuration level or the multiple interface level.

When an IPG is applied to a trunk group, it applies to all ports in the trunk group. When you are creating a new trunk group, the IPG setting on the primary port is automatically applied to the secondary ports.

Syntax: [no] ipg <value>

To configure an IPG of 112 on Ethernet interface 0/1/21, for example, enter the following command:

```
FGS624P Switch(config)#interface ethernet 0/1/21

FGS624P Switch(config-if-e1000-0/1/21)#ipg 112
```

or, for multiple interface levels, to configure IPG for ports 0/1/11 and 0/1/14 through 0/1/17, enter the following commands:

```
FGS624P Switch(config)#interface ethernet 0/1/11 e 0/1/14 to 0/1/17

FGS624P Switch(config-mif-0/1/11,0/1/14-0/1/17)#ipg 104
```

As a result of this configuration, the output from the show interface e 0/1/21 command is:

```
FGS624P Switch#show interfaces ethernet 0/1/21
GigabitEthernet0/1/21 is up, line protocol is up
  Hardware is GigabitEthernet, address is 00e0.5204.4014 (bia 00e0.5204.4014)
  Configured speed auto, actual 100Mbit, configured duplex fdx, actual fdx
  Configured mdi mode AUTO, actual MDIX
  Member of L2 VLAN ID 1, port is untagged, port state is FORWARDING
  BPDU Guard is disabled, Root Protect is disabled
  STP configured to ON, priority is level0
  Flow Control is config enabled, oper enabled, negotiation disabled
  Mirror disabled, Monitor disabled
  Not member of any active trunks
  Not member of any configured trunks
  No port name
  Inter-Packet Gap (IPG) is 112 bit times
  IP MTU 10222 bytes
  300 second input rate: 0 bits/sec, 0 packets/sec, 0.00% utilization
  300 second output rate: 248 bits/sec, 0 packets/sec, 0.00% utilization
  0 packets input, 0 bytes, 0 no buffer
  Received 0 broadcasts, 0 multicasts, 0 unicasts
  0 input errors, 0 CRC, 0 frame, 0 ignored
  0 runts, 0 giants
  80 packets output, 5120 bytes, 0 underruns
  Transmitted 0 broadcasts, 80 multicasts, 0 unicasts
  0 output errors, 0 collisions
```

This feature is supported on 10/100/1000M ports.

Enabling and Disabling Support for 100BaseFX

Some Foundry FastIron devices support 100BaseFX fiber transceivers. After you physically install a 100BaseFX transceiver, you must enter a CLI command to enable it.

NOTE: The CLI syntax for enabling and disabling 100BaseFX support on a Compact device differs from the syntax for a chassis device. Follow the appropriate instructions below.

Compact Device

This section shows how to enable 100BaseFX on a Compact device.

The Foundry device supports the following types of SFPs for 100BaseFX:

- Multimode SFP – maximum distance is 2 kilometers
- Bidirectional singlemode SFP – maximum distance is 10 kilometers
- Long Reach (LR) – maximum distance is 40 kilometers (introduced in software release FSX 03.1.00)
- Intermediate Reach (IR) – maximum distance is 15 kilometers (introduced in software release FSX 03.1.00)

NOTE: Support for copper optics in not available on combo-ports.

NOTE: Connect the 100BaseFX fiber transceiver *after* configuring both sides of the link. Otherwise, the link could become unstable, fluctuating between up and down states.

To enable 100BaseFX on a fiber port, enter the following command at the Global CONFIG level of the CLI:

```
FastIron(config)#link-config gig fiber 100base-fx e 4
```

The above command enables 100BaseFX on port 4.

The following command enables 100BaseFX on ports 3 and 4

```
FastIron(config)#link-config gig fiber 100base-fx e 3 e 4
```

Syntax: [no] link-config gig fiber 100base-fx ethernet [<stacknum>/<slotnum>/]<portnum> ethernet [<stacknum>/<slotnum>/]<portnum>

The <stacknum> parameter is required on FastIron GS devices running software release 02.5.00 or later.

The <slotnum> parameter is required on chassis devices and on FastIron GS devices running software release 02.5.00 or later.

The <portnum> parameter is a valid port number.

You can specify one or two ethernet ports at a time, as shown in the above examples.

To disable 100BaseFX support on a fiber port, enter the **no** form of the command. Note that you must disable 100BaseFX support before inserting a different type of module In the same port. Otherwise, the device will not recognize traffic traversing the port.

FSX 100/1000 Interface Module

The FSX 100/1000 Interface module and 100BaseFX SFP was introduced in software release 02.4.00.

NOTE: The following procedure applies to the FSX 100/1000 Fiber interface module only. The CLI syntax for enabling and disabling 100BaseFX support on the FSX differs than on a Compact device. Make sure you refer to the appropriate procedures.

The FSX 100/1000 fiber interface module supports the following types of SFPs for 100BaseFX:

- Multimode SFP – maximum distance is 2 kilometers
- Bidirectional single mode SFP – maximum distance is 10 kilometers
- Long Reach (LR) – maximum distance is 40 kilometers (introduced in software release FSX 03.1.00)
- Intermediate Reach (IR) – maximum distance is 15 kilometers (introduced in software release FSX 03.1.00)

NOTE: Connect the 100BaseFX fiber transceiver *after* configuring both sides of the link. Otherwise, the link could become unstable, fluctuating between up and down states.

To enable support for 100BaseFX on an FSX fiber port, enter commands such as the following:

```
FastIron(config)#interface e 1/6
FastIron(config-if-1/6)#100-fx
```

The above commands enable 100BaseFX on port 6 in slot 1.

Syntax: [no] 100-fx

To disable 100BaseFX support on a fiber port, enter the **no** form of the command. Note that you must disable 100BaseFX support before inserting a different type of module In the same port. Otherwise, the device will not recognize traffic traversing the port.

Changing the Gigabit Fiber Negotiation Mode

The globally configured Gigabit negotiation mode is the default mode for all Gigabit fiber ports. You can override the globally configured default and set individual ports to the following:

- Negotiate-full-auto – The port first tries to perform a handshake with the other port to exchange capability information. If the other port does not respond to the handshake attempt, the port uses the manually configured configuration information (or the defaults if an administrator has not set the information). This is the default.
- Auto-Gigabit – The port tries to perform a handshake with the other port to exchange capability information.
- Negotiation-off – The port does not try to perform a handshake. Instead, the port uses configuration information manually configured by an administrator.

To change the mode for individual ports, enter commands such as the following:

```
FastIron(config)#int ethernet 1 to 4
FastIron(config-mif-1-4)#gig-default auto-gig
```

This command overrides the global setting and sets the negotiation mode to auto-Gigabit for ports 1 – 4.

Syntax: gig-default neg-full-auto | auto-gig | neg-off

Modifying Port Priority (QoS)

You can give preference to the inbound traffic on specific ports by changing the Quality of Service (QoS) level on those ports. For information and procedures, see the chapter “Configuring Quality of Service” on page 21-1.

Enabling Dynamic Configuration of Voice over IP (VoIP) Phones

Platform Support:

- FESX/FSX/FWSX devices running software release 02.2.00 and later

You can configure a FastIron device to automatically detect and re-configure a VoIP phone when it is physically moved from one port to another within the same device. To do so, you must configure a ***voice VLAN ID*** on the port to which the VoIP phone is connected. The software stores the voice VLAN ID in the port's database for retrieval by the VoIP phone.

The dynamic configuration of a VoIP phone works in conjunction with the VoiP phone's discovery process. Upon installation, and sometimes periodically, a VoIP phone will query the Foundry device for VoIP information and will advertise information about itself, such as, device ID, port ID, and platform. When the Foundry device receives the VoIP phone's query, it sends the voice VLAN ID in a reply packet back to the VoIP phone. The VoIP phone then configures itself within the voice VLAN.

As long as the port to which the VoIP phone is connected has a voice VLAN ID, the phone will configure itself into that voice VLAN. If you change the voice VLAN ID, the software will immediately send the new ID to the VoIP phone, and the VoIP phone will re-configure itself with the new voice VLAN.

Configuration Notes

- This feature works with any VoIP phone that:
  - Runs CDP
  - Sends a VoIP VLAN query message
  - Can configure its voice VLAN after receiving the VoIP VLAN reply
- Automatic configuration of a VoIP phone will not work if one of the following applies:
  - You do not configure a voice VLAN ID for a port with a VoIP phone
  - You remove the configured voice VLAN ID from a port without configuring a new one
  - You remove the port from the voice VLAN
- Make sure the port is able to intercept CDP packets (**cdp run** command).
- Some VoIP phones may require a reboot after configuring or re-configuring a voice VLAN ID. For example, if your VoIP phone queries for VLAN information only once upon boot up, you must reboot the VoIP phone before it can accept the VLAN configuration. If your phone is powered by a PoE device, you can reboot the phone by disabling then re-enabling the port.
- Foundry devices do not currently support Cisco 7970 VOIP phones.

Enabling Dynamic Configuration of a Voice over IP (VoIP) phone

You can create a voice VLAN ID for a port, or for a group of ports.

To create a voice VLAN ID for a port, enter commands such as the following:

```
FastIron(config)#interface e 2
FastIron(config-if-e1000-2)#voice-vlan 1001
```

To create a voice VLAN ID for a group of ports, enter commands such as the following:

```
FastIron(config)#interface e 1-8
FastIron(config-mif-1-8)#voice-vlan 1001
```

Syntax: [no] voice-vlan <voice-vlan-num>

where <voice-vlan-num> is a valid VLAN ID between 1 – 4095.

To remove a voice VLAN ID, use the **no** form of the command.

Viewing Voice VLAN Configurations

You can view the configuration of a voice VLAN for a particular port or for all ports.

To view the voice VLAN configuration for a port, specify the port number with the **show voice-vlan** command. The following example shows the command output results.

```
FastIron#show voice-vlan ethernet 2

Voice vlan ID for port 2: 1001
```

The following example shows the message that appears when the port does not have a configured voice VLAN.

```
FastIron#show voice-vlan ethernet 2

Voice vlan is not configured for port 2.
```

To view the voice VLAN for all ports, use the **show voice-vlan** command. The following example shows the command output results.

```
FastIron#show voice-vlan

Port ID          Voice-vlan
2                1001
8                150
15               200
```

Syntax: show voice-vlan ethernet [[<stacknum>/<slotnum>/]<portnum>]

If you specify an ethernet port, note the following:

The <stacknum> parameter is required on FastIron GS devices running software release 02.5.00 or later.

The <slotnum> parameter is required on chassis devices and on FastIron GS devices running software release 02.5.00 or later.

The <portnum> parameter is a valid port number.

Configuring Port Flap Dampening

Platform Support:

- FESX/FSX/FWSX devices running software release 03.0.00 and later
- FGS and FLS devices running software release 04.0.00 and later

Port Flap Dampening increases the resilience and availability of the network by limiting the number of port state transitions on an interface.

If the port link state toggles from up to down for a specified number of times within a specified period, the interface is physically disabled for the specified wait period. Once the wait period expires, the port’s link state is re-enabled. However, if the wait period is set to zero (0) seconds, the port’s link state will remain disabled until it is manually re-enabled.

Configuration Notes

- When a flap dampening port becomes a member of a trunk group, that port, as well as all other member ports of that trunk group, will inherit the primary port’s configuration. This means that the member ports will inherit the primary port’s flap dampening configuration, regardless of any previous configuration.
- The Foundry device counts the number of times a port’s link state toggles from "up to down", and not from "down to up".
- The sampling time or window (the time during which the specified toggle threshold can occur before the wait period is activated) is triggered when the first "up to down" transition occurs.
- "Up to down" transitions include UDLD-based toggles, as well as the physical link state.

Configuring Port Flap Dampening on an Interface

This feature is configured at the interface level.

```
FastIron(config)#interface ethernet 2/1
FastIron(config-if-e10000-2/1)#link-error-disable 10 3 10
```

Syntax: [no] link-error-disable <toggle-threshold> <sampling-time-in-sec> <wait-time-in-sec>

The <toggle-threshold> is the number of times a port's link state goes from up to down and down to up before the wait period is activated. The default is 0. Enter a valid value range from 1-50.

The <sampling-time-in-sec> is the amount of time during which the specified toggle threshold can occur before the wait period is activated. The default is 0 seconds. Enter 0 – 65535 seconds.

The <wait-time-in-sec> is the amount of time the port remains disabled (down) before it becomes enabled. Entering 0 – 65535 seconds; 0 indicates that the port will stay down until an administrative override occurs.

Configuring Port Flap Dampening on a Trunk

You can configure the port flap dampening feature on the primary port of a trunk using the **link-error-disable** command. Once configured on the primary port, the feature is enabled on all ports that are members of the trunk. You cannot configure port flap dampening on port members of the trunk.

Enter commands such as the following on the primary port of a trunk.

```
FastIron(config)#interface ethernet 2/1
FastIron(config-if-e10000-2/1)#link-error-disable 10 3 10
```

Re-enabling a Port Disabled by Port Flap Dampening

A port disabled by port flap dampening is automatically re-enabled once the wait period expires; however, if the wait period is set to zero (0) seconds, you must re-enable the port by entering the following command on the disabled port:

```
FastIron(config)#interface ethernet 2/1
FastIron(config-if-e10000-2/1)#no link-error-disable 10 3 10
```

Displaying Ports Configured with Port Flap Dampening

Ports that have been disabled due to the port flap dampening feature are identified in the output of the **show link-error-disable** command. The following shows an example output.

```
FastIron#show link-error-disable

Port 2/1 is forced down by link-error-disable.
```

Use the **show link-error-disable all** command to display the ports with the port flap dampening feature enabled.

For FSX software releases prior to release 03.2.00 and for FGS software releases, the output of the command shows the following:

```
FastIron#show link-error-disable all

Port8/1 is configured for link-error-disable
            threshold:1, sampling_period:10, waiting_period:0
Port8/2 is configured for link-error-disable
            threshold:1, sampling_period:10, waiting_period:0
Port8/3 is configured for link-error-disable
            threshold:1, sampling_period:10, waiting_period:0
Port8/4 is configured for link-error-disable
            threshold:1, sampling_period:10, waiting_period:0
Port8/5 is configured for link-error-disable
            threshold:4, sampling_period:10, waiting_period:2
Port8/9 is configured for link-error-disable
             threshold:2, sampling_period:20, waiting_period:0
```

For FESX/FSX/FWSX devices running software release 03.2.00, the output of the command shows the following:

```
FastIron#show link-error-disable all
 Port   -----------------Config---------------   ------Oper----
  #    Threshold  Sampling-Time  Shutoff-Time   State  Counter
-----   ---------  -------------  ------------   -----  -------
   11          3            120           600    Idle      N/A
   12          3            120           500    Down      424
```

Table 4.5 defines the port flap dampening statistics displayed by the **show link-error-disable all** command.

Table 4.5: Output of show link-error-disable

This Column...	Displays...
Port #	The port number.
Threshold	The number of times the port's link state will go from up to down and down to up before the wait period is activated.
Sampling-Time	The number of seconds during which the specified toggle threshold can occur before the wait period is activated.
Shutoff-Time	The number of seconds the port will remain disabled (down) before it becomes enabled. A zero (0) indicates that the port will stay down until an administrative override occurs.
State	The port's state can be one of the following: • **Idle** – The link is normal and no link state toggles have been detected or sampled. • **Down** – The port is disabled because the number of sampled errors exceeded the configured threshold. • **Err** – The port sampled one or more errors.

Table 4.5: Output of show link-error-disable

This Column...	Displays...
Counter	• If the port's state is **Idle**, this field displays **N/A**. • If the port's state is **Down**, this field shows the remaining value of the shutoff timer. • If the port's state is **Err**, this field shows the number of errors sampled.

Syntax: show link-error-disable [all]

Also in FESX/FSX/FWSX devices running software release 03.2.00 and later, the **show interface** command indicates if the port flap dampening feature is enabled on the port. For example:

```
FastIron#show interface ethernet 15
GigabitEthernet15 is up, line protocol is up
  Link Error Dampening is Enabled
  Hardware is GigabitEthernet, address is 00e0.5200.010e (bia 00e0.5200.010e)
  Configured speed auto, actual 1Gbit, configured duplex fdx, actual fdx
  Configured mdi mode AUTO, actual MDIX

FastIron#show interface ethernet 17
GigabitEthernet17 is ERR-DISABLED, line protocol is down
  Link Error Dampening is Enabled
  Hardware is GigabitEthernet, address is 00e0.5200.010e (bia 00e0.5200.010e)
  Configured speed auto, actual unknown, configured duplex fdx, actual unknown
```

The line "Link Error Dampening" displays "Enabled" if port flap dampening is enabled on the port or "Disabled" if the feature is disabled on the port. The feature is enabled on the ports in the two examples above. Also, the characters "ERR-DISABLED" is displayed for the "GigabitEthernet" line if the port is disabled because of link errors.

Syntax: show interface ethernet <port-number>

In addition to the show commands above, the output of the **show interface brief** command for devices running FESX/FSX/FWSX devices running software release 03.2.00 indicates if a port is down due to link errors. For example:

```
FastIron#show interface brief e17

Port  Link    State     Dupl Speed Trunk Tag Priori MAC            Name
17    ERR-DIS None      None None  15    Yes level0 00e0.5200.010e
```

The ERR-DIS entry under the "Link" column indicates the port is down due to link errors.

Syslog Messages for Port Flap Dampening

The following Syslog messages are generated on devices running FSX software release 03.2.00 and later.

- If the threshold for the number of times that a port's link toggles from "up" to "down" then "down" to "up" has been exceeded, the following Syslog message is displayed:

  ```
  0d00h02m10s:I:ERR_DISABLE: Link flaps on port ethernet 16 exceeded threshold;
  port in err-disable state
  ```

- If the wait time (port is down) expires and the port is brought up the following Syslog message is displayed:

  ```
  0d00h02m41s:I:ERR_DISABLE: Interface ethernet 16, err-disable recovery timeout
  ```

Configuring a Local MAC Address for Layer 2 Management Traffic

Platform Support:

- FGS and FLS devices running software release 04.0.00 and later

By default, Foundry Layer 2 devices use the MAC address of the first port as the MAC address for Layer 2 management traffic. For example, when the Foundry device receives an ARP request for its management IP address, it responds with the first port's MAC address. This may cause problems in some configurations where the Foundry device uses the same MAC address for management traffic as for switched traffic.

Starting with the software releases listed above, you can configure the Foundry device to use a different MAC address for Layer 2 management traffic than for switched traffic. When you issue the **use-local-management-mac**, the Foundry device changes a local bit in the first port's MAC address and uses this MAC address for management traffic. The second bit of the first port's MAC address is changed to 2. For example, if the MAC address is 00e0.5201.9900 after the feature is enabled, the switch uses 02e0.5201.9900 for management functions. Switched traffic will continue to use the first port's MAC address without the local bit setting.

EXAMPLE:

```
FastIron(config)#use-local-management-mac
FastIron(config)#write memory
FastIron(config)#end
FastIron#reload
```

Syntax: [no] use-local-management-mac

NOTE: You must save the configuration and reload the software to place the change into effect.

NOTE: This feature is only available for the switch code. It is not available for router code.

Port Loop Detection

Platform Support:

- FGS/FLS devices running software release 04.0.00 and later

This feature allows the Foundry device to disable a port that is on the receiving end of a loop by sending test packets. You can configure the time period during which test packets are sent.

Strict Mode and Loose Mode

There are two types of loop detection; Strict Mode and Loose Mode. In Strict Mode, a port is disabled only if a packet is looped back to that same port. Strict Mode overcomes specific hardware issues where packets are echoed back to the input port. In Strict Mode, loop detection must be configured on the physical port.

In Loose Mode, loop detection is configured on the VLAN of the receiving port. Loose Mode disables the receiving port if packets originate from any port or VLAN on the same device. The VLAN of the receiving port must be configured for loop detection in order to disable the port.

Recovering Disabled Ports

Once a loop is detected on a port, it is placed in Err-Disable state. The port will remain disabled until one of the following occurs:

- You manually disable and enable the port at the Interface Level of the CLI.
- You enter the command **clear loop-detection**. This command clears loop detection statistics and enables all Err-Disabled ports.
- The device automatically re-enables the port. To set your device to automatically re-enable Err-Disabled ports, see “Configuring the Device to Automatically Re-Enable Ports” .

Configuration Notes

The following information applies to Loose Mode loop detection:

- With Loose Mode, two ports of a loop are disabled.
- Different VLANs may disable different ports. A disabled port affects every VLAN using it.
- Loose Mode floods test packets to the entire VLAN. This can impact system performance if too many VLANs are configured for Loose Mode loop detection.

NOTE: Foundry recommends that you limit the use of Loose Mode. If you have a large number of VLANS, configuring loop detection on all of them can significantly affect system performance because of the flooding of test packets to all configured VLANs. An alternative to configuring loop detection in a VLAN-group of many VLANs is to configure a separate VLAN with the same tagged port and configuration, and enable loop detection on this VLAN only.

NOTE: When loop detection is used with L2 loop prevention protocols, such as spanning tree (STP), the L2 protocol takes higher priority. Loop detection cannot send or receive probe packets if ports are blocked by L2 protocols, so it doesn't detect L2 loops when STP is running because loops within a VLAN have been prevented by STP. Loop detection running in Loose Mode can detect and break L3 loops because STP cannot prevent loops across different VLANs. In these instances, the ports are not blocked and loop detection is able to send out probe packets in one VLAN and receive packets in another VLAN. In this way, loop detection running in Loose Mode disables both ingress and egress ports.

Enabling Loop Detection

Use the **loop-detection** command to enable loop detection on a physical port (Strict Mode) or a VLAN (Loose Mode). Loop detection is disabled by default. The following example shows a Strict Mode configuration.

```
FastIron(config)#interface ethernet 1/1
FastIron(config-if-e1000-1/1)#loop-detection
```

The following example shows a Loose Mode configuration:

```
FastIron(config)#vlan20
FastIron(config-vlan-20)#loop-detection
```

By default, the port will send test packets every one second, or the number of seconds specified by the **loop-detection-interval** command. See “Configuring a Global Loop Detection Interval” on page 4-35.

Syntax: [no] loop-detection

Use the [no] form of the command to disable loop detection.

Configuring a Global Loop Detection Interval

The loop detection interval specifies how often a test packet is sent on a port. When loop detection is enabled, the loop detection time unit is 0.1 second, with a default of 10 (one second). The range is from 1 (one tenth of a second) to 100 (10 seconds). You can use the **show loop-detection status** command to view the loop detection interval.

To configure the global loop detection interval, enter a command similar to the following:

```
FastIron(config)#loop-detection-interval 50
```

This command sets the loop-detection interval to 5 seconds (50 x 0.1).

To revert to the default global loop detection interval of 10, enter one of the following:

```
FastIron(config)#loop-detection-interval 10
```

OR

```
FastIron(config)#no loop-detection-interval 50
```

Syntax: [no] loop-detection-interval <number>

where <number> is a value from 1 to 100. The system multiplies your entry by 0.1 to calculate the interval at which test packets will be sent.

Configuring the Device to Automatically Re-Enable Ports

To configure the Foundry device to automatically re-enable ports that were disabled because of a loop detection, enter the following command:

```
FastIron(config)#errdisable recovery cause loop-detection
```

The above command will cause the Foundry device to automatically re-enable ports that were disabled because of a loop detection. By default, the device will wait 300 seconds before re-enabling the ports. You can optionally change this interval to a value from 10 to 65535 seconds. See “Specifying the Recovery Time Interval” on page 4-36.

Syntax: [no] errdisable recovery cause loop-detection

Use the [no] form of the command to disable this feature.

Specifying the Recovery Time Interval

The recovery time interval specifies the number of seconds the Foundry device will wait before automatically re-enabling ports that were disabled because of a loop detection. (See also “Configuring the Device to Automatically Re-Enable Ports” on page 4-36.) By default, the device will wait 300 seconds. To change the recovery time interval, enter a command such as the following:

```
FastIron(config)#errdisable recovery interval 120
```

The above command configures the device to wait 120 seconds (2 minutes) before re-enabling the ports.

To revert back to the default recovery time interval of 300 seconds (5 minutes), enter one of the following commands:

```
FastIron(config)#errdisable recovery interval 300
```

OR

```
FastIron(config)#no errdisable recovery interval 120
```

Syntax: [no] errdisable recovery interval <seconds>

where <seconds> is a number from 10 to 65535.

Clearing Loop-Detection

To clear loop detection statistics and re-enable all ports that are in Err-Disable state because of a loop detection, enter the following command:

```
FastIron#clear loop-detection
```

Displaying Loop-Detection Information

Use the **show loop-detection status** command to display loop detection status, as shown:

```
FastIron#show loop-detection status
loop detection packets interval: 10 (unit 0.1 sec)
Number of err-disabled ports: 3
You can re-enable err-disable ports one by one by "disable" then "enable"
under interface config, re-enable all by "clear loop-detect", or
configure "errdisable recovery cause loop-detection" for automatic recovery
index port/vlan  status                        #errdis  sent-pkts recv-pkts
1       1/13     untag, LEARNING               0         0         0
2       1/15     untag, BLOCKING               0         0         0
3       1/17     untag, DISABLED               0         0         0
4       1/18     ERR-DISABLE by itself         1         6         1
5       1/19     ERR-DISABLE by vlan 12        0         0         0
6     vlan12     2 ERR-DISABLE ports           2         24        2
```

If a port is errdisabled in Strict mode, it shows “ERR-DISABLE by itself”. If it is errdisabled due to its associated vlan, it shows “ERR-DISABLE by vlan ?”

The following command displays the current disabled ports, including the cause and the time:

```
FastIron#show loop-detection disable
Number of err-disabled ports: 3
You can re-enable err-disable ports one by one by "disable" then "enable"
under interface config, re-enable all by "clear loop-detect", or
configure "errdisable recovery cause loop-detection" for automatic recovery
index  port         caused-by    disabled-time
1      1/18         itself       00:13:30
2      1/19         vlan 12      00:13:30
3      1/20         vlan 12      00:13:30
```

This example shows the disabled ports, the cause, and the time the port was disabled. If loop-detection is configured on a physical port, the disable cause will show “itself”. For VLANs configured for loop-detection, the cause will be a VLAN.

The following command shows the hardware and software resources being used by the loop-detection feature:

```
Vlans configured loop-detection use 1 HW MAC
Vlans not configured but use HW MAC: 1 10

                      alloc in-use  avail get-fail    limit  get-mem  size init
configuration pool       16      6     10        0     3712        6    15   16
linklist pool            16     10      6        0     3712       10    16   16
```

Syslog Message

The following message is logged when a port is disabled due to loop detection. This message also appears on the console:

```
loop-detect: port ?\?\? vlan ?, into errdisable state
```

The Errdisable function logs a message whenever it re-enables a port.

Chapter 5

Operations, Administration, and Maintenance

This chapter describes how to perform management, administration, and maintenance operations on FastIron devices. These operations include software image management, upgrade management, and scheduling system maintenance tasks.

Overview

For easy software image management, all Foundry devices support the download and upload of software images between the flash modules on the devices and a Trivial File Transfer Protocol (TFTP) server on the network.

Foundry devices have two flash memory modules:

- ***Primary flash*** – The default local storage device for image files and configuration files.
- ***Secondary flash*** – A second flash storage device. You can use the secondary flash to store redundant images for additional booting reliability or to preserve one software image while testing another one.

Only one flash device is active at a time. By default, the primary image will become active upon reload.

You can update the software contained on a flash module using TFTP to copy the update image from a TFTP server onto the flash module. In addition, you can copy software images and configuration files from a flash module to a TFTP server.

NOTE: Foundry devices are TFTP clients but not TFTP servers. You must perform the TFTP transaction from the Foundry device. You cannot “put” a file onto the Foundry device using the interface of your TFTP server.

NOTE: If you are attempting to transfer a file using TFTP but have received an error message, see “Diagnostic Error Codes and Remedies for TFTP Transfers” on page 5-21.

Determining the Software Versions Installed and Running on a Device

Use the following methods to display the software versions running on the device and the versions installed in flash memory.

Determining the Flash Image Version Running on the Device

To determine the flash image version running on a device, enter the **show version** command at any level of the CLI. Some examples are shown below.

Compact Devices

To determine the flash image version running on a Compact device, enter the **show version** command at any level of the CLI. The following shows an example output.

```
FastIron#show version
  SW: Version 03.0.00T53 Copyright (c) 1996-2002 Foundry Networks, Inc.
      Compiled on Mar 26 2003 at 13:50:31 labeled as FER03000
      (3089381 bytes) from Primary fer03000.bin
  HW: Stackable FES2402-PREM-ILP
==========================================================================
  330 MHz Power PC processor 8245 (version 129/1014) 66 MHz bus
  512 KB boot flash memory
16384 KB code flash memory
  128 MB DRAM
Monitor Option is on
The system uptime is 4 days 4 hours 8 minutes 33 seconds
The system : started=warm start
```

The version information is shown in bold type in this example.

- “03.0.00T53” indicates the flash code version number. The “T53” is used by Foundry for record keeping.
- “labeled as FER03000” indicates the flash code image label. The label indicates the image type and version and is especially useful if you change the image file name.
- “Primary fer03000.bin” indicates the flash code image file name that was loaded.

Chassis Devices

To determine the flash image version running on a chassis device, enter the **show version** command at any level of the CLI. The following is an example output.

```
FastIron Switch#show version
==========================================================================
Active Management CPU:
SW: Version 03.1.00aT3e3 Copyright (c) 1996-2006 Foundry Networks, Inc.
Compiled on Nov 07 2006 at 10:20:07 labeled as SXR03100a
(3613675 bytes) from Primary sxr03100a.bin
BootROM: Version 03.0.01T3e5 (FEv2)
HW: Chassis FastIron SX 1600-PREM
Serial #: TE15065544
==========================================================================
Standby Management CPU:
SW: Version 03.1.00aT3e3 Copyright (c) 1996-2006 Foundry Networks, Inc.
Compiled on Nov 07 2006 at 10:20:07 labeled as SXR03100a
BootROM: Version 03.0.01T3e5 (FEv2)
==========================================================================
SL 1: SX-F424C 24-port Gig Copper
Serial #: CH03060022
P-ASIC 0: type 00D1, rev D2
P-ASIC 1: type 00D1, rev D2
==========================================================================
SL 5: SX-F42XG 2-port 10G
Serial #: CH19050324
P-ASIC 8: type 01D1, rev 00
P-ASIC 9: type 01D1, rev 00
==========================================================================
SL 9: SX-FIZMR4 0-port Management
Serial #: Non-exist
==========================================================================
SL 10: SX-FIZMR4 0-port Management
Serial #: Non-exist
==========================================================================
SL 13: SX-F424C 24-port Gig Copper
Serial #: Non-exist
P-ASIC 24: type 00D1, rev D2
P-ASIC 25: type 00D1, rev D2
==========================================================================
SL 18: SX-F42XG 2-port 10G
Serial #: CH13050374
P-ASIC 34: type 01D1, rev 00
P-ASIC 35: type 01D1, rev 00
==========================================================================
Active Management Module:
660 MHz Power PC processor 8541 (version 32/0020) 66 MHz bus
512 KB boot flash memory
16384 KB code flash memory
512 MB DRAM
Standby Management Module:
660 MHz Power PC processor 8541 (version 32/0020) 66 MHz bus
512 KB boot flash memory
16384 KB code flash memory
512 MB DRAM
The system uptime is 2 days 4 hours 33 minutes 52 seconds
The system : started=warm start reloaded=by "reload"
```

The version information is shown in bold type in this example.

- “03.1.00aT3e3” indicates the flash code version number. The “T3e3” is used by Foundry for record keeping.

- “labeled as SXR03100a” indicates the flash code image label. The label indicates the image type and version and is especially useful if you change the image file name.
- “Primary SXR03100a.bin” indicates the flash code image file name that was loaded.

Determining the Boot Image Version Running on the Device

To determine the boot image running on a device, enter the **show flash** command at any level of the CLI. The following shows an example output.

```
FastIron#sh flash
Active Management Module (Slot 9):
Compressed Pri Code size = 3613675, Version 03.1.00aT3e3 (sxr03100a.bin)
Compressed Sec Code size = 2250218, Version 03.1.00aT3e1 (sxs03100a.bin)
Compressed BootROM Code size = 524288, Version 03.0.01T3e5
Code Flash Free Space = 9699328
Standby Management Module (Slot 10):
Compressed Pri Code size = 3613675, Version 03.1.00aT3e3 (sxr03100a.bin)
Compressed Sec Code size = 2250218, Version 03.1.00aT3e1 (sxs03100a.bin)
Compressed BootROM Code size = 524288, Version 03.0.01T3e5
Code Flash Free Space = 524288
```

The boot code version is shown in bold type.

Determining the Image Versions Installed in Flash Memory

Enter the **show flash** command to display the boot and flash images installed on the device. An example of the command’s output is shown in “Determining the Boot Image Version Running on the Device” .

- The “Compressed Pri Code size” line lists the flash code version installed in the primary flash area.
- The “Compressed Sec Code size” line lists the flash code version installed in the secondary flash area.
- The “Boot Monitor Image size” line lists the boot code version installed in flash memory. The device does not have separate primary and secondary flash areas for the boot image. The flash memory module contains only one boot image.

NOTE: To minimize the boot-monitor image size, **ping** and **tftp** operations performed in the boot-monitor mode are restricted to copper ports on the FastIron Chassis management modules and to copper ports on the FastIron stackable swtich combination copper and fiber ports. The fiber ports on these modules do not have the ability to **ping** or **tftp** from the boot-monitor mode.

Flash Image Verification

The Flash Image Verification feature allows you to verify boot images based on hash codes, and to generate hash codes where needed. This feature lets you select from three data integrity verification algorithms:

- MD5 - Message Digest algorithm (RFC 1321)
- SHA1 - US Secure Hash Algorithm (RFC 3174)
- CRC - Cyclic Redundancy Checksum algorithm

CLI Commands

Use the following command syntax to verify the flash image:

Syntax: verify md5 | sha1 | crc32 <ASCII string> | primary | secondary [<hash code>]

Platform Support:

- FGS and FLS devices running software release 04.0.00 and later

  - FESX/FSX/FWSX devices running software release 04.0.00 and later
- **md5** – Generates a 16-byte hash code
- **sha1** – Generates a 20-byte hash code
- **crc32** – Generates a 4 byte checksum
- **ascii string –** A valid image filename
- **primary** – The primary boot image (primary.img)
- **secondary** – The secondary boot image (secondary.img)
- **hash code** – The hash code to verify

The following examples show how the **verify** command can be used in a variety of circumstances:

To generate an MD5 hash value for the secondary image, enter the following command:

```
FastIron#verify md5 secondary
FastIron#.........................Done
Size = 2044830, MD5 01c410d6d153189a4a5d36c955653862
```

To generate a SHA-1 hash value for the secondary image, enter the following command:

```
FastIron#verify sha secondary
FastIron#.........................Done
Size = 2044830, SHA1 49d12d26552072337f7f5fcaef4cf4b742a9f525
```

To generate a CRC32 hash value for the secondary image, enter the following command:

```
FastIron#verify crc32 secondary
FastIron#.........................Done
Size = 2044830, CRC32 b31fcbc0
```

To verify the hash value of a secondary image with a known value, enter the following commands:

```
FastIron#verify md5 secondary 01c410d6d153189a4a5d36c955653861
FastIron#.........................Done
Size = 2044830, MD5 01c410d6d153189a4a5d36c955653862
Verification FAILED.
```

In the previous example, the codes did not match, and verification failed. If verification succeeds, the output will look like this:

```
FastIron#verify md5 secondary 01c410d6d153189a4a5d36c955653861
FastIron#.........................Done
Size = 2044830, MD5 01c410d6d153189a4a5d36c955653861
Verification SUCEEDED.
```

The following examples show this process for SHA-1 and CRC32 algorithms:

```
FastIron#verify sha secondary 49d12d26552072337f7f5fcaef4cf4b742a9f525
FastIron#.........................Done
Size = 2044830, sha 49d12d26552072337f7f5fcaef4cf4b742a9f525
Verification SUCCEEDED.
```

and:

```
FastIron#verify crc32 secondary b31fcbc0
FastIron#.........................Done
Size = 2044830, CRC32 b31fcbc0
Verification SUCCEEDED.
```

Image File Types

This section lists the boot and flash image file types supported on the FastIron family of switches and how to install them. For information about a specific version of code, see the release notes.

Table 5.1: Software Image Files

Product	Boot Image[1]	Flash Image
FESX pre-release 02.3.01	FEXZ*xxxxx*.bin	FEXS*xxxxx*.bin
FESX release 02.3.01 through pre-release 03.0.00	FEXZ*xxxxx*.bin	SXS*xxxxx*.bin (Layer 2) or SXL*xxxxx*.bin (Base Layer 3) or SXR*xxxxx*.bin (Full Layer 3)
FESX release 03.0.00 and later	SXZ*xxxxx*.bin	SXS*xxxxx*.bin (Layer 2) or SXL*xxxxx*.bin (Base Layer 3) or SXR*xxxxx*.bin (Full Layer 3)
FGS and FLS	FGZ*xxxxx*.bin	FGS*xxxxx*.bin (Layer 2) or FGL*xxxxx*.bin (Base Layer 3)
FSX, FSX 800, and FSX 1600	SXZ*xxxxx*.bin	SXS*xxxxx*.bin (Layer 2) or SXL*xxxxx*.bin (Base Layer 3) or SXR*xxxxx*.bin (Full Layer 3)
FWSX	FWXZ*xxxxx*.bin	FWXS*xxxxx*.bin (Layer 2)

1. These images are applicable to these devices only and are not interchangeable. For example, you cannot load FWSX boot or flash images on a FSX device, and vice versa. Also, you cannot load other images, such as B2R or B2S, for BigIron devices, on the FastIron family of switches.

Upgrading Software

Use the following procedures to upgrade the software.

NOTE: This section does not describe how to upgrade a FESX or FSX base model to a premium (PREM) model. To perform this upgrade, you need an upgrade kit. Contact Foundry Networks for information.

Note Regarding Upgrading from FGS 02.4.00 to the New Release

When upgrading to release 02.5.00, the software automatically converts the saved system configuration to the new stack/slot/port nomenclature (see “Port Nomenclature on the FastIron GS and FastIron LS” on page 2-3). The software makes the configuration change only for the saved configuration when the device is started.

Software release 02.5.00 is not backward-compatible. If the software is downgraded from release 02.5.00, the configuration must be reloaded.

Migrating to the New Release (FESX and FSX devices only)

Beginning with release 02.3.01, FESX and FSX devices share the same flash images. In releases prior to 02.3.01, FESX and FSX flash images were separate and were issued via separate software releases. Starting with release 02.3.01, the flash images for these devices were merged and are now issued in the same software release.

The new, combined flash images may create unique software upgrade circumstances for FESX and FSX devices. (FWSX devices are not affected by the software merge.) If your device is currently running software release 02.2.00 or later (FESX devices), or 02.2.01a or later (FSX devices), your device is not affected by the software merge. However, if your FESX or FSX device is running a release earlier than these versions, you must first upgrade the software on your device to FESX release 02.2.00 or later, or FSX release 02.2.01a or later, *before* loading the new software image. Earlier releases will not allow you to load the 02.3.01 or later software image.

To determine which software version is running on your device, use the **show version** command.

See the following sections for information on how to upgrade the software images on your device.

Upgrading from FESX pre-02.2.00 or FSX pre-02.2.01a to the New Release

If your device is running a software release earlier than FESX 02.2.00 or FSX 02.2.01a, you must first upgrade it to FESX 02.2.00 or later, or FSX 02.2.01a or later, before you can upgrade it to the new release. Follow the instructions, below.

1. Upgrade your device to software release FESX 02.2.00 or later, or FSX 02.2.01a or later. Follow the steps presented in “Upgrading Software” on page 5-6 and “Upgrading the Flash Code” on page 5-8. Make sure you reload the software after loading the flash code.
2. Upgrade your device to the new software release. Refer to one of the following sections:
    - FESX – “Upgrading from FESX 02.2.00 or later to the New Release” on page 5-7.
    - FSX – “Upgrading from FSX 02.2.01a or later to the New Release” on page 5-7.

Upgrading from FESX 02.2.00 or later to the New Release

1. Upgrade the boot code to the new version (FEXZ0*xxxx*.bin) using the steps presented in “Upgrading Software” on page 5-6.
2. Upgrade the flash code to the new version using the steps presented in “Upgrading the Flash Code” on page 5-8.

Upgrading from FSX 02.2.01a or later to the New Release

1. Upgrade the boot code to the new version (SXZ0*xxxx*.bin) using the steps presented in “Upgrading Software” on page 5-6.
2. Upgrade the flash code to the new version using the steps presented in “Upgrading the Flash Code” on page 5-8.

Upgrading the Boot Code

NOTE:

- If you are upgrading a FESX or FSX device, see “Migrating to the New Release (FESX and FSX devices only)” on page 5-7 before performing the steps in this section.
- If you are upgrading an FGS device from release 02.4.00, see “Note Regarding Upgrading from FGS 02.4.00 to the New Release” on page 5-6 before performing the steps in this section.

1. Place the new boot code on a TFTP server to which the Foundry device has access.
2. Enter the following command at the Privileged EXEC level of the CLI (example: `FastIron Switch#`) to copy the boot code from the TFTP server into flash memory:
   - **copy tftp flash** <ip-addr> <image-file-name> **bootrom**
3. Verify that the code has been successfully copied by entering the following command at any level of the CLI:
   - **show flash**

   The output will display the compressed boot ROM code size and the boot code version.
4. Upgrade the flash code as instructed in the following section.

Upgrading the Flash Code

NOTE: If you are upgrading a FESX or FSX device, see "Migrating to the New Release (FESX and FSX devices only)" on page 5-7 before performing the steps in this section.

1. Place the new flash code on a TFTP server to which the Foundry device has access.
2. Enter the following command at the Privileged EXEC level of the CLI (example: `FastIron Switch#`) to copy the flash code from the TFTP server into the flash memory:
   - **copy tftp flash** <ip-addr> <image-file-name> **primary | secondary**
3. Verify that the flash code has been successfully copied by entering the following command at any level of the CLI:
   - **show flash**
4. If the flash code version is correct, go to Step 5. Otherwise, go to Step 1.
5. Reload the software by entering one of the following commands:
   - **reload** (this command boots from the default boot source, which is the primary flash area by default)
   - **boot system flash primary | secondary**

NOTE: Release 03.1.00a added a confirmation step in the boot system flash process. This step occurs after a boot system flash primary/secondary command is entered and gives an administrator the opportunity to make last minute changes or corrections before performing a reload. The example below shows the confirmation step:

```
FastIron Switch#boot system flash primary
Are you sure? (enter ‘Y’ or ‘N’): y
```

Boot Code Synchronization Feature

Release 03.1.00a added support for automatic synchronization of the boot image in the active and redundant management modules. When the new boot image is copied into the active module, it is automatically synchronized with the redundant management module.

NOTE: There is currently no option for manual synchronization of the boot image.

To activate the boot synchronization process, enter the following command:

```
FastIron#copy tftp flash 192.168.255.102 superx/boot/sxz03001.bin bootrom
```

The system responds with the following message:

```
FastIron#Load to buffer (8192 bytes per dot)
..................Write to boot flash......................
TFTP to Flash Done.
```

```
FastIron#Synchronizing with standby module...
Boot image synchronization done.
```

Using SNMP to Upgrade Software

You can use a third-party SNMP management application such as HP OpenView to upgrade software on a Foundry device.

NOTE: The syntax shown in this section assumes that you have installed HP OpenView in the "/usr" directory.

NOTE: Foundry recommends that you make a backup copy of the startup-config file before you upgrade the software. If you need to run an older release, you will need to use the backup copy of the startup-config file.

1. Configure a read-write community string on the Foundry device, if one is not already configured. To configure a read-write community string, enter the following command from the global CONFIG level of the CLI:

   snmp-server community <string> **ro | rw**

   where <string> is the community string and can be up to 32 characters long.
2. On the Foundry device, enter the following command from the global CONFIG level of the CLI:

   no snmp-server pw-check

   This command disables password checking for SNMP set requests. If a third-party SNMP management application does not add a password to the password field when it sends SNMP set requests to a Foundry device, by default the Foundry device rejects the request.
3. From the command prompt in the UNIX shell, enter the following command:

   /usr/OV/bin/snmpset -c <rw-community-string> <fdry-ip-addr> **1.3.6.1.4.1.1991.1.1.2.1.5.0 ipaddress** <tftp-ip-addr> **1.3.6.1.4.1.1991.1.1.2.1.6.0 octetstringascii** <file-name> **1.3.6.1.4.1.1991.1.1.2.1.7.0 integer** <command-integer>

   where:

   <rw-community-string> is a read-write community string configured on the Foundry device.

   <fdry-ip-addr> is the Foundry device's IP address.

   <tftp-ip-addr> is the TFTP server's IP address.

   <file-name> is the image file name.

   <command-integer> is one of the following:

   20 – Download the flash code into the device's primary flash area.

   22 – Download the flash code into the device's secondary flash area.

Changing the Block Size for TFTP File Transfers

When you use TFTP to copy a file to or from a Foundry device, the device transfers the data in blocks of 8192 bytes by default. You can change the block size to one of the following if needed:

- 4096
- 2048
- 1024
- 512

- 256
- 128
- 64
- 32
- 16

To change the block size for TFTP file transfers, enter a command such as the following at the global CONFIG level of the CLI:

```
FastIron(config)#flash 2047
set flash copy block size to 2048
```

Syntax: [no] flash <num>

The software rounds up the <num> value you enter to the next valid power of two, and displays the resulting value. In this example, the software rounds the value up to 2048.

NOTE: If the value you enter is one of the valid powers of two for this parameter, the software still rounds the value up to the next valid power of two. Thus, if you enter 2048, the software rounds the value up to 4096.

Rebooting

You can use boot commands to immediately initiate software boots from a software image stored in primary or secondary flash on a Foundry device or from a BootP or TFTP server. You can test new versions of code on a Foundry device or choose the preferred boot source from the console boot prompt without requiring a system reset.

NOTE: It is very important that you verify a successful TFTP transfer of the boot code ***before*** you reset the system. If the boot code is not transferred successfully but you try to reset the system, the system will not have the boot code with which to successfully boot.

By default, the Foundry device first attempts to boot from the image stored in its primary flash, then its secondary flash, and then from a TFTP server. You can modify this booting sequence at the global CONFIG level of the CLI using the **boot system…** command.

To initiate an immediate boot from the CLI, enter one of the **boot system…** commands.

NOTE: In FESX/FSX/FWSX devices running software release 03.0.00 and higher, the **boot system tftp** command is supported on ports e 1 through e 12 only.

Displaying the Boot Preference

Use the **show boot-preference** command to display the boot sequence in the startup config and running config files. The boot sequence displayed is also identified as either user-configured or the default.

The following example shows the default boot sequence preference.

```
FastIron#show boot-preference

Boot system preference (Configured):
   Use Default

Boot system preference(Default):
   Boot system flash primary
   Boot system flash secondary
```

The following example shows a user-configured boot sequence preference:

```
FastIron#show boot-preference

Boot system preference(Configured):
   Boot system flash secondary
   Boot system tftp 10.1.1.1 FGS04000b1.bin
   Boot system flash primary

Boot system preference (Default)
   Boot system flash primary
   Boot system flash secondary
```

Syntax: show boot-preference

Platform Support:

- FESX/FSX/FWSX devices running software release 04.0.00
- FGS and FLS devices running software release 04.0.00

The results of the **show run** command for the configured example above appear as follows:

```
FastIron#show run

Current Configuration:
!
ver 04.0.00x1T7el
!
module 1 fgs-48-port-copper-base-module
module 2 fgs-xfp-1-port-10g-module
module 3 fgs-xfp-1-port-10g-module
!
alias cp=copy tf 10.1.1.1 FGS04000bl.bin pri
!
!
boot sys fl sec
boot sys df 10.1.1.1 FGS04000bl.bin
boot sys fl pri
ip address 10.1.1.4 255.255.255.0
snmp-client 10.1.1.1
!
end
```

Loading and Saving Configuration Files

For easy configuration management, all Foundry devices support both the download and upload of configuration files between the devices and a TFTP server on the network.

You can upload either the startup configuration file or the running configuration file to the TFTP server for backup and use in booting the system.

- ***Startup configuration file*** – This file contains the configuration information that is currently saved in flash. To display this file, enter the **show configuration** command at any CLI prompt.
- ***Running configuration file*** – This file contains the configuration active in the system RAM but not yet saved to flash. These changes could represent a short-term requirement or general configuration change. To display this file, enter the **show running-config** or **write terminal** command at any CLI prompt.

Each device can have one startup configuration file and one running configuration file. The startup configuration file is shared by both flash modules. The running configuration file resides in DRAM.

When you load the startup-config file, the CLI parses the file three times.

1. During the first pass, the parser searches for **system-max** commands. A **system-max** command changes the size of statically configured memory.
2. During the second pass, the parser implements the **system-max** commands if present and also implements trunk configuration commands (**trunk** command) if present.
3. During the third pass, the parser implements the remaining commands.

Replacing the Startup Configuration with the Running Configuration

After you make configuration changes to the active system, you can save those changes by writing them to flash memory. When you write configuration changes to flash memory, you replace the startup configuration with the running configuration.

To replace the startup configuration with the running configuration, enter the following command at any Enable or CONFIG command prompt:

```
FastIron#write memory
```

Replacing the Running Configuration with the Startup Configuration

If you want to back out of the changes you have made to the running configuration and return to the startup configuration, enter the following command at the Privileged EXEC level of the CLI:

```
FastIron#reload
```

Logging Changes to the Startup-Config File

You can configure a Foundry device to generate a Syslog message when the startup-config file is changed. The trap is enabled by default.

The following Syslog message is generated when the startup-config file is changed:

```
startup-config was changed
```

If the startup-config file was modified by a valid user, the following Syslog message is generated:

```
startup-config was changed by <username>
```

To disable or re-enable Syslog messages when the startup-config file is changed, use the following command:

Syntax: [no] logging enable config-changed

Copying a Configuration File to or from a TFTP Server

To copy the startup-config or running-config file to or from a TFTP server, use one of the following methods.

NOTE: You can name the configuration file when you copy it to a TFTP server. However, when you copy a configuration file from the server to a Foundry device, the file is always copied as “startup-config” or “running-config”, depending on which type of file you saved to the server.

To initiate transfers of configuration files to or from a TFTP server using the CLI, enter one of the following commands:

- **copy startup-config tftp** <tftp-ip-addr> <filename> – Use this command to upload a copy of the startup configuration file from the Layer 2 Switch or Layer 3 Switch to a TFTP server.
- **copy running-config tftp** <tftp-ip-addr> <filename> – Use this command to upload a copy of the running configuration file from the Layer 2 Switch or Layer 3 Switch to a TFTP server.
- **copy tftp startup-config** <tftp-ip-addr> <filename> – Use this command to download a copy of the startup configuration file from a TFTP server to a Layer 2 Switch or Layer 3 Switch.

Dynamic Configuration Loading

You can load dynamic configuration commands (commands that do not require a reload to take effect) from a file on a TFTP server into a Foundry device’s running-config. You can make configuration changes off-line, then load the changes directly into the device’s running-config, without reloading the software.

Usage Considerations

- Use this feature only to load configuration information that does not require a software reload to take effect. For example, you cannot use this feature to change statically configured memory (**system-max** command) or to enter trunk group configuration information into the running-config.
- Do not use this feature if you have deleted a trunk group but have not yet placed the changes into effect by saving the configuration and then reloading. When you delete a trunk group, the command to configure the trunk group is removed from the device’s running-config, but the trunk group remains active. To finish deleting a trunk group, save the configuration (to the startup-config file), then reload the software. After you reload the software, then you can load the configuration from the file.
- Do not load port configuration information for secondary ports in a trunk group. Since all ports in a trunk group use the port configuration settings of the primary port in the group, the software cannot implement the changes to the secondary port.

Preparing the Configuration File

A configuration file that you create must follow the same syntax rules as the startup-config file the device creates.

- The configuration file is a script containing CLI configuration commands. The CLI reacts to each command entered from the file in the same way the CLI reacts to the command if you enter it. For example, if the command results in an error message or a change to the CLI configuration level, the software responds by displaying the message or changing the CLI level.
- The software retains the running-config that is currently on the device, and changes the running-config only by adding new commands from the configuration file. If the running config already contains a command that is also in the configuration file you are loading, the CLI rejects the new command as a duplicate and displays an error message. For example, if the running-config already contains a a command that configures ACL 1, the software rejects ACL 1 in the configuration file, and displays a message that ACL 1 is already configured.
- The file can contain global CONFIG commands or configuration commands for interfaces, routing protocols, and so on. You cannot enter User EXEC or Privileged EXEC commands.
- The default CLI configuration level in a configuration file is the global CONFIG level. Thus, the first command in the file must be a global CONFIG command or “ ! ”. The ! (exclamation point) character means “return to the global CONFIG level”.

  NOTE: You can enter text following “ ! “ as a comment. However, the “ !” is not a comment marker. It returns the CLI to the global configuration level.

NOTE: If you copy-and-paste a configuration into a management session, the CLI ignores the " ! " instead of changing the CLI to the global CONFIG level. As a result, you might get different results if you copy-and-paste a configuration instead of loading the configuration using TFTP.

- Make sure you enter each command at the correct CLI level. Since some commands have identical forms at both the global CONFIG level and individual configuration levels, if the CLI's response to the configuration file results in the CLI entering a configuration level you did not intend, then you can get unexpected results.

  For example, if a trunk group is active on the device, and the configuration file contains a command to disable STP on one of the secondary ports in the trunk group, the CLI rejects the commands to enter the interface configuration level for the port and moves on to the next command in the file you are loading. If the next command is a spanning-tree command whose syntax is valid at the global CONFIG level as well as the interface configuration level, then the software applies the command globally. Here is an example:

  The configuration file contains these commands:

  ```
  interface ethernet 2
  no spanning-tree
  ```

  The CLI responds like this:

  ```
  FastIron(config)#interface ethernet 2
  Error - cannot configure secondary ports of a trunk
  FastIron(config)#no spanning-tree
  FastIron(config)#
  ```

- If the file contains commands that must be entered in a specific order, the commands must appear in the file in the required order. For example, if you want to use the file to replace an IP address on an interface, you must first remove the old address using "no" in front of the **ip address** command, then add the new address. Otherwise, the CLI displays an error message and does not implement the command. Here is an example:

  The configuration file contains these commands:

  ```
  interface ethernet 11
  ip address 10.10.10.69/24
  ```

  The running-config already has a command to add an address to port 11, so the CLI responds like this:

  ```
  FastIron(config)#interface ethernet 11
  FastIron(config-if-e1000-11)#ip add 10.10.10.69/24
  Error: can only assign one primary ip address per subnet
  FastIron(config-if-e1000-11)#
  ```

  To successfully replace the address, enter commands into the file as follows:

  ```
  interface ethernet 11
  no ip address 20.20.20.69/24
  ip address 10.10.10.69/24
  ```

  This time, the CLI accepts the command, and no error message is displayed:

  ```
  FastIron(config)#interface ethernet 11
  FastIron(config-if-e1000-11)#no ip add 20.20.20.69/24
  FastIron(config-if-e1000-111)#ip add 10.10.10.69/24
  FastIron(config-if-e1000-11)
  ```

- Always use the **end** command at the end of the file. The **end** command must appear on the last line of the file, by itself.

Loading the Configuration Information into the Running-Config

To load the file from a TFTP server, use either of the following commands:

- **copy tftp running-config** <ip-addr> <filename>
- **ncopy tftp** <ip-addr> <filename> **running-config**

NOTE: If you are loading a configuration file that uses a truncated form of the CLI command **access-list**, the software will not go into batch mode.

For example, the following command line *will* initiate batch mode:

```
access-list 131 permit host pc1 host pc2
```

The following command line *will not* initiate batch mode:

```
acc 131 permit host pc1 host pc2
```

Maximum File Sizes for Startup-Config File and Running-Config

Each Foundry device has a maximum allowable size for the running-config and the startup-config file. If you use TFTP to load additional information into a device's running-config or startup-config file, it is possible to exceed the maximum allowable size. If this occurs, you will not be able to save the configuration changes.

The maximum size for the running-config and the startup-config file is 64K each.

To determine the size of a Foundry device's running-config or startup-config file, copy it to a TFTP server, then use the directory services on the server to list the size of the copied file. To copy the running-config or startup-config file to a TFTP server, use one of the following commands.

- Commands to copy the running-config to a TFTP server:
  - **copy running-config tftp** <ip-addr> <filename>
  - **ncopy running-config tftp** <ip-addr> <from-name>
- Commands to copy the startup-config file to a TFTP server:
  - **copy startup-config tftp** <ip-addr> <filename>
  - **ncopy startup-config tftp** <ip-addr> <from-name>

Loading and Saving Configuration Files with IPv6

Platform Support:

- FESX/FSX/FWSX devices running software release 02.4.00 and later
- FGS and FLS devices running software release 04.0.00 and later

The **copy** command for IPv6 allows you to do the following:

- Copy a file from a specified source to an IPv6 TFTP server
- Copy a file from an IPv6 TFTP server to a specified destination

Copying a File to an IPv6 TFTP Server

You can copy a file from the following sources to an IPv6 TFTP server:

- Flash memory
- Running configuration
- Startup configuration

Copying a File from Flash Memory

For example, to copy the primary or secondary boot image from the device's flash memory to an IPv6 TFTP server, enter a command such as the following:

```
FastIron#copy flash tftp 2001:7382:e0ff:7837::3 test.img secondary
```

This command copies the secondary boot image named test.img from flash memory to a TFTP server with the IPv6 address of 2001:7382:e0ff:7837::3.

Syntax: copy flash tftp <ipv6-address> <source-file-name> primary | secondary

The <ipv6-address> parameter specifies the address of the TFTP server. You must specify this address in hexadecimal using 16-bit values between colons as documented in RFC 2373.

The <source-file-name> parameter specifies the name of the file you want to copy to the IPv6 TFTP server.

The **primary** keyword specifies the primary boot image, while the **secondary** keyword specifies the secondary boot image.

Copying a File from the Running or Startup Configuration

For example, to copy the running configuration to an IPv6 TFTP server, enter a command such as the following:

```
FastIron#copy running-config tftp 2001:7382:e0ff:7837::3 newrun.cfg
```

This command copies the running configuration to a TFTP server with the IPv6 address of 2001:7382:e0ff:7837::3 and names the file on the TFTP server newrun.cfg.

Syntax: copy running-config | startup-config tftp <ipv6-address> <destination-file-name>

Specify the **running-config** keyword to copy the running configuration file to the specified IPv6 TFTP server.

Specify the **startup-config** keyword to copy the startup configuration file to the specified IPv6 TFTP server.

The tftp <ipv6-address> parameter specifies the address of the TFTP server. You must specify this address in hexadecimal using 16-bit values between colons as documented in RFC 2373.

The <destination-file-name> parameter specifies the name of the file that is copied to the IPv6 TFTP server.

Copying a File from an IPv6 TFTP Server

You can copy a file from an IPv6 TFTP server to the following destinations:

- Flash memory
- Running configuration
- Startup configuration

Copying a File to Flash Memory

For example, to copy a boot image from an IPv6 TFTP server to the primary or secondary storage location in the device’s flash memory, enter a command such as the following:

```
FastIron#copy tftp flash 2001:7382:e0ff:7837::3 test.img secondary
```

This command copies a boot image named test.img from an IPv6 TFTP server with the IPv6 address of 2001:7382:e0ff:7837::3 to the secondary storage location in the device’s flash memory.

Syntax: copy tftp flash <ipv6-address> <source-file-name> primary | secondary

The <ipv6-address> parameter specifies the address of the TFTP server. You must specify this address in hexadecimal using 16-bit values between colons as documented in RFC 2373.

The <source-file-name> parameter specifies the name of the file you want to copy from the IPv6 TFTP server.

The **primary** keyword specifies the primary storage location in the device’s flash memory, while the **secondary** keyword specifies the secondary storage location in the device’s flash memory.

Copying a File to the Running or Startup Configuration

For example, to copy a configuration file from an IPv6 TFTP server to the router’s running or startup configuration, enter a command such as the following.

```
FastIron#copy tftp running-config 2001:7382:e0ff:7837::3 newrun.cfg overwrite
```

This command copies the newrun.cfg file from the IPv6 TFTP server and overwrites the router’s running configuration file with the contents of newrun.cfg.

NOTE: To activate this configuration, you must reload (reset) the device.

Syntax: copy tftp running-config | startup-config <ipv6-address> <source-file-name> [overwrite]

Specify the **running-config** keyword to copy the running configuration from the specified IPv6 TFTP server.

Specify the **startup-config** keyword to copy the startup configuration from the specified IPv6 TFTP server.

The <ipv6-address> parameter specifies the address of the TFTP server. You must specify this address in hexadecimal using 16-bit values between colons as documented in RFC 2373.

The <source-file-name> parameter specifies the name of the file that is copied from the IPv6 TFTP server.

The **overwrite** keyword specifies that the device should overwrite the current configuration file with the copied file. If you do not specify this parameter, the device copies the file into the current running or startup configuration but does not overwrite the current configuration.

Using the IPv6 Ncopy Command

Platform Support:

- FESX/FSX/FWSX devices running software release 02.4.00 and later
- FGS and FLS devices running software release 04.0.00 and later

The **ncopy** command for IPv6 allows you to do the following:

- Copy a primary or secondary boot image from flash memory to an IPv6 TFTP server.
- Copy the running configuration to an IPv6 TFTP server.
- Copy the startup configuration to an IPv6 TFTP server
- Upload various files from an IPv6 TFTP server.

Copying a Primary or Secondary Boot Image from Flash Memory to an IPv6 TFTP Server

For example, to copy the primary or secondary boot image from the device's flash memory to an IPv6 TFTP server, enter a command such as the following:

```
FastIron#ncopy flash primary tftp 2001:7382:e0ff:7837::3 primary.img
```

This command copies the primary boot image named primary.img from flash memory to a TFTP server with the IPv6 address of 2001:7382:e0ff:7837::3.

Syntax: ncopy flash primary | secondary tftp <ipv6-address> <source-file-name>

The **primary** keyword specifies the primary boot image, while the **secondary** keyword specifies the secondary boot image.

The **tftp** <ipv6-address> parameter specifies the address of the TFTP server. You must specify this address in hexadecimal using 16-bit values between colons as documented in RFC 2373.

The <source-file-name> parameter specifies the name of the file you want to copy from flash memory.

Copying the Running or Startup Configuration to an IPv6 TFTP Server

For example, to copy a device's running or startup configuration to an IPv6 TFTP server, enter a command such as the following:

```
FastIron#ncopy running-config tftp 2001:7382:e0ff:7837::3 bakrun.cfg
```

This command copies a device's running configuration to a TFTP server with the IPv6 address of 2001:7382:e0ff:7837::3 and names the destination file bakrun.cfg.

Syntax: ncopy running-config | startup-config tftp <ipv6-address> <destination-file-name>

Specify the **running-config** keyword to copy the device's running configuration or the **startup-config** keyword to copy the device's startup configuration.

The **tftp** <ipv6-address> parameter specifies the address of the TFTP server. You must specify this address in hexadecimal using 16-bit values between colons as documented in RFC 2373.

The <destination-file-name> parameter specifies the name of the running configuration that is copied to the IPv6 TFTP server.

Uploading Files from an IPv6 TFTP Server

You can upload the following files from an IPv6 TFTP server:

- Primary boot image.
- Secondary boot image.
- Running configuration.
- Startup configuration.

Uploading a Primary or Secondary Boot Image from an IPv6 TFTP Server

For example, to upload a primary or secondary boot image from an IPv6 TFTP server to a device's flash memory, enter a command such as the following:

```
FastIron#ncopy tftp 2001:7382:e0ff:7837::3 primary.img flash primary
```

This command uploads the primary boot image named primary.img from a TFTP server with the IPv6 address of 2001:7382:e0ff:7837::3 to the device's primary storage location in flash memory.

Syntax: ncopy tftp <ipv6-address> <source-file-name> flash primary | secondary

The **tftp** <ipv6-address> parameter specifies the address of the TFTP server. You must specify this address in hexadecimal using 16-bit values between colons as documented in RFC 2373.

The <source-file-name> parameter specifies the name of the file you want to copy from the TFTP server.

The **primary** keyword specifies the primary location in flash memory, while the **secondary** keyword specifies the secondary location in flash memory.

Uploading a Running or Startup Configuration from an IPv6 TFTP Server

For example to upload a running or startup configuration from an IPv6 TFTP server to a device, enter a command such as the following:

```
FastIron#ncopy tftp 2001:7382:e0ff:7837::3 newrun.cfg running-config
```

This command uploads a file named newrun.cfg from a TFTP server with the IPv6 address of 2001:7382:e0ff:7837::3 to the device.

Syntax: ncopy tftp <ipv6-address> <source-file-name> running-config | startup-config

The **tftp** <ipv6-address> parameter specifies the address of the TFTP server. You must specify this address in hexadecimal using 16-bit values between colons as documented in RFC 2373.

The <source-file-name> parameter specifies the name of the file you want to copy from the TFTP server.

Specify the **running-config** keyword to upload the specified file from the IPv6 TFTP server to the device. The device copies the specified file into the current running configuration but does not overwrite the current configuration.

Specify the **startup-config** keyword to upload the specified file from the IPv6 TFTP server to the device. The the device copies the specified file into the current startup configuration but does not overwrite the current configuration.

Using SNMP to Save and Load Configuration Information

You can use a third-party SNMP management application such as HP OpenView to save and load a Foundry device's configuration. To save and load configuration information using HP OpenView, use the following procedure.

NOTE: The syntax shown in this section assumes that you have installed HP OpenView in the "/usr" directory.

1. Configure a read-write community string on the Foundry device, if one is not already configured. To configure a read-write community string, enter the following command from the global CONFIG level of the CLI:

   snmp-server community <string> **ro | rw**

   where <string> is the community string and can be up to 32 characters long.

2. On the Foundry device, enter the following command from the global CONFIG level of the CLI:

   no snmp-server pw-check

   This command disables password checking for SNMP set requests. If a third-party SNMP management application does not add a password to the password field when it sends SNMP set requests to a Foundry device, by default the Foundry device rejects the request.

3. From the command prompt in the UNIX shell, enter the following command:

   /usr/OV/bin/snmpset -c <rw-community-string> <fdry-ip-addr> **1.3.6.1.4.1.1991.1.1.2.1.5.0 ipaddress** <tftp-ip-addr> **1.3.6.1.4.1.1991.1.1.2.1.8.0 octetstringascii** <config-file-name> **1.3.6.1.4.1.1991.1.1.2.1.9.0 integer** <command-integer>

   where:

   <rw-community-string> is a read-write community string configured on the Foundry device.

   <fdry-ip-addr> is the Foundry device's IP address.

   <tftp-ip-addr> is the TFTP server's IP address.

   <config-file-name> is the configuration file name.

   <command-integer> is one of the following:

   20 – Upload the startup-config file from the Foundry device's flash memory to the TFTP server.

   21 – Download a startup-config file from a TFTP server to the Foundry device's flash memory.

   22 – Upload the running-config from the Foundry device's flash memory to the TFTP server.

   23 – Download a configuration file from a TFTP server into the Foundry device's running-config.

   NOTE: Option **23** adds configuration information to the running-config on the device, and does not replace commands. If you want to replace configuration information in the device, use "no" forms of the configuration commands to remove the configuration information, then use configuration commands to create the configuration information you want. Follow the guidelines in "Dynamic Configuration Loading" on page 5-13.

Erasing Image and Configuration Files

To erase software images or configuration files, use the commands described below. These commands are valid at the Privileged EXEC level of the CLI.

- **erase flash primary** erases the image stored in primary flash of the system.
- **erase flash secondary** erases the image stored in secondary flash of the system.
- **erase startup-config** erases the configuration stored in the startup configuration file; however, the running configuration remains intact until system reboot.

Scheduling a System Reload

In addition to reloading the system manually, you can configure the Foundry device to reload itself at a specific time or after a specific amount of time has passed.

NOTE: The scheduled reload feature requires the system clock. You can use a Simple Network Time Protocol (SNTP) server to set the clock or you can set the device clock manually. See "Specifying a Simple Network Time Protocol (SNTP) Server" on page 4-9 or "Setting the System Clock" on page 4-11.

Reloading at a Specific Time

To schedule a system reload for a specific time, use the **reload at** command. For example, to schedule a system reload from the primary flash module for 6:00:00 AM, April 1, 2003, enter the following command at the global CONFIG level of the CLI:

```
FastIron#reload at 06:00:00 04-01-03
```

Syntax: reload at <hh:mm:ss> <mm-dd-yy> [primary | secondary]

<hh:mm:ss> is the hours, minutes, and seconds.

<mm-dd-yy> is the month, day, and year.

primary | secondary specifies whether the reload is to occur from the primary code flash module or the secondary code flash module. The default is **primary**.

Reloading after a Specific Amount of Time

To schedule a system reload to occur after a specific amount of time has passed on the system clock, use **reload after** command. For example, to schedule a system reload from the secondary flash one day and 12 hours later, enter the following command at the global CONFIG level of the CLI:

```
FastIron#reload after 01:12:00 secondary
```

Syntax: reload after <dd:hh:mm> [primary | secondary]

<dd:hh:mm> is the number of days, hours, and minutes.

primary | secondary specifies whether the reload is to occur from the primary code flash module or the secondary code flash module.

Displaying the Amount of Time Remaining Before a Scheduled Reload

To display how much time is remaining before a scheduled system reload, enter the following command from any level of the CLI:

```
FastIron#show reload
```

Canceling a Scheduled Reload

To cancel a scheduled system reload using the CLI, enter the following command at the global CONFIG level of the CLI:

```
FastIron#reload cancel
```

Diagnostic Error Codes and Remedies for TFTP Transfers

If an error occurs with a TFTP transfer to or from a Foundry Layer 2 Switch or Layer 3 Switch, one of the following error codes displays on the console.

Error code	Message	Explanation and action
1	Flash read preparation failed.	A flash error occurred during the download. Retry the download. If it fails again, contact customer support.
2	Flash read failed.	
3	Flash write preparation failed.	
4	Flash write failed.	
5	TFTP session timeout.	TFTP failed because of a time out. Check IP connectivity and make sure the TFTP server is running.
6	TFTP out of buffer space.	The file is larger than the amount of room on the device or TFTP server. If you are copying an image file to flash, first copy the other image to your TFTP server, then delete it from flash. (Use the **erase flash...** CLI command at the Privileged EXEC level to erase the image in the flash.) If you are copying a configuration file to flash, edit the file to remove unneeded information, then try again.
7	TFTP busy, only one TFTP session can be active.	Another TFTP transfer is active on another CLI session, or Web management session, or IronView Network Manager session. Wait, then retry the transfer.
8	File type check failed.	You accidentally attempted to copy the incorrect image code into the system. For example, you might have tried to copy a Chassis image into a Compact device. Retry the transfer using the correct image.
16	TFTP remote - general error.	The TFTP configuration has an error. The specific error message describes the error. Correct the error, then retry the transfer.
17	TFTP remote - no such file.	
18	TFTP remote - access violation.	
19	TFTP remote - disk full.	
20	TFTP remote - illegal operation.	
21	TFTP remote - unknown transfer ID.	
22	TFTP remote - file already exists.	
23	TFTP remote - no such user.	

Chapter 6

IPv6 Management on the FastIron GS and FastIron LS

This chapter describes the IPv6 management features available for FastIron GS and FastIron LS devices, including command syntax and management examples.

NOTE: For details about IPv6 management on FastIron X Series devices, see “IPv6 Host Support” on page 28-12

IPv6 Management Overview

IPv6 was designed to replace IPv4, the Internet protocol that is most commonly used currently throughout the world. IPv6 increases the number of network address bits from 32 (IPv4) to 128 bits, which provides more than enough unique IP addresses to support all of the network devices on the planet into the future. IPv6 is expected to quickly become the network standard.

Foundry FastIron devices that support IPv6 may be used as management hosts. Interfaces on these devices are configured with IPv6 addresses, but do not have full IPv6 routing enabled. IPv6 is available on all FastIron devices that are running Layer 2, Base Layer 3, or Full Layer 3 software images.

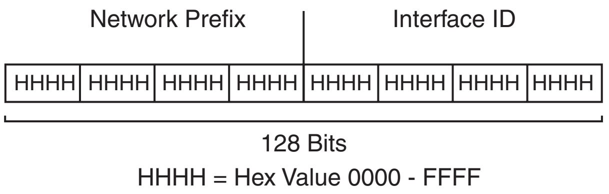
NOTE: Foundry FastIron devices can serve as management hosts on an IPv6 network. However, IPv6 ***routing*** functionality is not supported for these devices.

IPv6 Addressing

IPv4 is limited because of the 32-bit addressing format, which cannot satisfy potential increases in the number of users, geographical needs, and emerging applications. To address this limitation, IPv6 introduces a new 128-bit addressing format.

An IPv6 address is composed of 8 fields of 16-bit hexadecimal values separated by colons (:). Figure 6.1 shows the IPv6 address format.

Figure 6.1 IPv6 Address Format


Network Prefix
Interface ID
HHHH HHHH HHHH HHHH HHHH HHHH HHHH HHHH
128 Bits
HHHH = Hex Value 0000 - FFFF

As shown in Figure 6.1, HHHH is a 16-bit hexadecimal value, while H is a 4-bit hexadecimal value. The following is an example of an IPv6 address:

2001:0000:0000:0200:002D:D0FF:FE48:4672

Note that this IPv6 address includes hexadecimal fields of zeros. To make the address less cumbersome, you can do the following:

- Omit the leading zeros; for example, 2001:0:0:200:2D:D0FF:FE48:4672.
- Compress the successive groups of zeros at the beginning, middle, or end of an IPv6 address to two colons (::) once per address; for example, 2001::200:2D:D0FF:FE48:4672.

When specifying an IPv6 address in a command syntax, keep the following in mind:

- You can use the two colons (::) only once in the address to represent the longest successive hexadecimal fields of zeros
- The hexadecimal letters in IPv6 addresses are not case-sensitive

As shown in Figure 6.1, the IPv6 network prefix is composed of the left-most bits of the address. As with an IPv4 address, you can specify the IPv6 prefix using the <prefix>/<prefix-length> format, where the following applies:

The <prefix> parameter is specified as 16-bit hexadecimal values separated by a colon.

The <prefix-length> parameter is specified as a decimal value that indicates the left-most bits of the IPv6 address.

The following is an example of an IPv6 prefix:

2001:FF08:49EA:D088::/64

Enabling and Disabling IPv6 on FastIron Devices

IPv6 is enabled by default for Foundry devices that support it. If desired, you can disable IPv6 on a global basis on an device by entering the following command at the Global CONFIG level of the CLI:

```
FastIron(config)#no ipv6 enable
```

Syntax: no ipv6 enable

To re-enable IPv6 after it has been disabled, enter the **ipv6 enable** command.

IPv6 Management Features

Platform Support:

- FGS and FLS devices running software release 04.0.00 and later

This section describes the CLI management commands that are available to FastIron devices that support IPv6.

IPv6 Access List

Platform Support:

- FGS and FLS devices running software release 04.0.00 and later

When you enter the **ipv6 access-list** command, the Foundry device enters the IPv6 Access List configuration level, where you can access several commands for configuring IPv6 ACL entries. .

NOTE: Unlike IPv4, there is no distinction between standard and extended ACLs in IPv6.

EXAMPLES:

```
FastIron(config)#ipv6 access-list netw
FastIron(config-ipv6-access-list-netw)#
```

Syntax: [no] ipv6 access-list <acl name>

The <acl name> parameter specifies a name for the IPv6 ACL. An IPv6 ACL name cannot start with a numeral, for example, 1access. Also, an IPv4 ACL and an IPv6 ACL cannot share the same name.

Configuring an IPv6 ACL

You can configure an IPv6 ACL to filter traffic to or from the IPv6 host, as shown in the following example:

EXAMPLES:

The following example blocks Telnet traffic received on port 0/1/1 from IPv6 host 2000:2182:e0bb::2:

```
FastIron(config)#ipv6 access-list fdry
FastIron(config-ipv6-access-list-fdry)#deny tcp host 2000:2382:e0bb::2/64 any eq
telnet
FastIron(config-ipv6-access-list-fdry)#permit ipv6 any any
```

Syntax: ipv6 access-list <name> deny | permit <ipv6-source-prefix>/<prefix-length> | any <ipv6-destination-prefix>/<prefix-length> | any [sequence <number>]

The **any** keyword, when specified instead of the <ipv6-source-prefix>/<prefix-length> or <ipv6-destination-prefix>/<prefix-length> parameters, matches any IPv6 prefix and is equivalent to the IPv6 prefix ::/0.

The **host** <ipv6-source-address> and **host** <ipv6-destination-address> parameter lets you specify a host IPv6 address. When you use this parameter, you do not need to specify the prefix length. A prefix length of 128 is implied.

The <name> parameter specifies a name for the IPv6 ACL. An IPv6 ACL name cannot start with a numeral, for example, 1access. Also, an IPv4 ACL and an IPv6 ACL cannot share the same name.

The **deny** keyword specifies that the request from the remote host is denied if it matches the specified source and destination prefixes.

The **permit** keyword specifies that the request from the remote host is permitted if it matches the specified source and destination prefixes.

The <ipv6-source-prefix>/<prefix-length> and <ipv6-destination-prefix>/<prefix-length> parameters specify a source or destination prefix and prefix length that a request must match for the specified action (deny or permit) to occur. You must specify the <ipv6-source-prefix> or <ipv6-destination-prefix> parameter in hexadecimal using 16-bit values between colons as documented in RFC 2373. You must specify the <prefix-length> parameter as a decimal value. A slash mark (/) must follow the <ipv6-prefix> parameter and precede the <prefix-length> parameter.

Table 6.1 describes the syntax arguments for IPv6 ACLs.

Table 6.1:Syntax Descriptions

Arguments...	Description...
ipv6 access-list <acl name>	Enables the IPv6 configuration level and defines the name of the IPv6 ACL. The <acl name> can contain up to 199 characters and numbers, but cannot begin with a number and cannot contain any spaces or quotation marks.
permit	The ACL will permit (forward) packets that match a policy in the access list.
deny	The ACL will deny (drop) packets that match a policy in the access list.
icmp	Indicates the you are filtering ICMP packets.
protocol	The type of IPv6 packet you are filtering. You can specify a well-known name for some protocols whose number is less than 255. For other protocols, you must enter the number. Enter "?" instead of a protocol to list the well-known names recognized by the CLI. IPv6 protocols include: • **AHP** – Authentication Header • **ESP** – Encapsulating Security Payload • **IPv6** – Internet Protocol version 6 • **SCTP** – Stream Control Transmission Protocol
<ipv6-source-prefix>/<prefix-length>	The <ipv6-source-prefix>/<prefix-length> parameter specify a source prefix and prefix length that a packet must match for the specified action (deny or permit) to occur. You must specify the <ipv6-source-prefix> parameter in hexadecimal using 16-bit values between colons as documented in RFC 2373. You must specify the <prefix-length> parameter as a decimal value. A slash mark (/) must follow the <ipv6-prefix> parameter and precede the <prefix-length> parameter.
<ipv6-destination-prefix>/<prefix-length>	The <ipv6-destination-prefix>/<prefix-length> parameter specify a destination prefix and prefix length that a packet must match for the specified action (deny or permit) to occur. You must specify the <ipv6-destination-prefix> parameter in hexadecimal using 16-bit values between colons as documented in RFC 2373. You must specify the <prefix-length> parameter as a decimal value. A slash mark (/) must follow the <ipv6-prefix> parameter and precede the <prefix-length> parameter
any	When specified instead of the <ipv6-source-prefix>/<prefix-length> or <ipv6-destination-prefix>/<prefix-length> parameters, matches any IPv6 prefix and is equivalent to the IPv6 prefix::/0.
host	Allows you specify a host IPv6 address. When you use this parameter, you do not need to specify the prefix length. A prefix length of all128 is implied.
icmp-type	ICMP packets can be filtered by ICMP message type. The type is a number from 0 to 255.

Table 6.1:Syntax Descriptions

Arguments...	Description...
icmp code	ICMP packets, which are filtered by ICMP message type can also be filtered by the ICMP message code. The code is a number from 0 to 255,
icmp-message	ICMP packets are filtered by ICMP messages.
tcp	Indicates the you are filtering TCP packets.
udp	Indicates the you are filtering UDP packets.
<ipv6-source-prefix>/<prefix-length>	The <ipv6-source-prefix>/<prefix-length> parameter specify a source prefix and prefix length that a packet must match for the specified action (deny or permit) to occur. You must specify the <ipv6-source-prefix> parameter in hexadecimal using 16-bit values between colons as documented in RFC 2373. You must specify the <prefix-length> parameter as a decimal value. A slash mark (/) must follow the <ipv6-prefix> parameter and precede the <prefix-length> parameter.
<ipv6-destination-prefix>/<prefix-length>	The <ipv6-destination-prefix>/<prefix-length> parameter specify a destination prefix and prefix length that a packet must match for the specified action (deny or permit) to occur. You must specify the <ipv6-destination-prefix> parameter in hexadecimal using 16-bit values between colons as documented in RFC 2373. You must specify the <prefix-length> parameter as a decimal value. A slash mark (/) must follow the <ipv6-prefix> parameter and precede the <prefix-length> parameter
any	When specified instead of the <ipv6-source-prefix>/<prefix-length> or <ipv6-destination-prefix>/<prefix-length> parameters, matches any IPv6 prefix and is equivalent to the IPv6 prefix::/0.
host	Allows you specify a host IPv6 address. When you use this parameter, you do not need to specify the prefix length. A prefix length of all128 is implied.

Table 6.1:Syntax Descriptions

Arguments...	Description...
tcp-udp-operator	The <tcp-udp-operator> parameter can be one of the following: • **eq** – The policy applies to the TCP or UDP port name or number you enter after **eq**. • **gt** – The policy applies to TCP or UDP port numbers greater than the port number or the numeric equivalent of the port name you enter after **gt**. Enter "?" to list the port names. • **lt** – The policy applies to TCP or UDP port numbers that are less than the port number or the numeric equivalent of the port name you enter after **lt**. • **neq** – The policy applies to all TCP or UDP port numbers except the port number or port name you enter after **neq**. • **range** – The policy applies to all TCP port numbers that are between the first TCP or UDP port name or number and the second one you enter following the **range** parameter. The range includes the port names or numbers you enter. For example, to apply the policy to all ports between and including 23 (Telnet) and 53 (DNS), enter the following: **range 23 53**. The first port number in the range must be lower than the last number in the range. The <source-port number> and <destination-port-number> for the tcp-udp-operator is the number of the port.
ipv6-operator	Allows you to filter the packets further by using one of the following options: • **dscp** – The policy applies to packets that match the traffic class value in the traffic class field of the IPv6 packet header. This operator allows you to filter traffic based on TOS or IP precedence. You can specify a value from 0 – 63. • **fragments** – The policy applies to fragmented packets that contain a non-zero fragment offset. **NOTE:** This option is not applicable to filtering based on source or destination port, TCP flags, and ICMP flags. • **routing** – The policy applies only to IPv6 source-routed packets. **NOTE:** This option is not applicable to filtering based on source or destination port, TCP flags, and ICMP flags. •
802.1p-priority-matching <number>	If you want to match only those packets that have the same 802.1p priorities as specified in the ACL. Enter 0 – 7.
dscp-marking <number>	Use the **dscp-marking** <number> parameter to specify a new QoS value to the packet. If a packet matches the filters in the ACL statement, this parameter assigns the DSCP value that you specify to the packet. Enter 0 – 63.

Table 6.1:Syntax Descriptions

Arguments...	Description...
802.1p-priority-marking <number>	Use the **802.1p-priority-marking** <number> parameter to specify a new QoS value to the packet. If a packet matches the filters in the ACL statement, this parameter assigns the 802.1p priority that you specify to the packet. Enter 0 – 7.
internal-priority-marking <number>	Use the **internal-priority-marking** <number> parameter to specify a new QoS value to the packet. If a packet matches the filters in the ACL statement, this parameter assigns the internal priority that you specify to the packet. Enter 0 – 7.
dscp-marking <number>	Use the **dscp-marking** <number> **dscp-cos-mapping** parameters parameters to specify a DSCP value and map that value to an internal QoS table to obtain the packet's new QoS value. The following occurs when you use these parameters. • You enter 0 – 63 for the **dscp-marking** <number> parameter. • The **dscp-cos-mapping** parameter takes the DSCP value you specified and compares it to an internal QoS table, which is indexed by DSCP values. The corresponding 802.1p priority, internal forwarding priority, and DSCP value is assigned to the packet.
dscp-cos-mapping	Use **dscp-cos-mapping** if you want to use the DSCP value in the packet's header to alter its QoS value. When you enter **dscp-cos-mapping**, the DSCP value in the packet's header is compared to a column in the internal QoS table. The 802.1p priority, internal forwarding priority, and DSCP value that are mapped to the matching column is assigned to the packet.

IPv6 Debug

Platform Support:

- FGS and FLS devices running software release 04.0.00 and later

The **debug ipv6** commands enable the collection of information about IPv6 configurations for troubleshooting.

Syntax: debug ipv6 <address> <cache> <icmp> <mld> <nd> <packet> <ra>

- address - IPv6 address
- cache - IPv6 cache entry
- icmp - ICMPv6
- mld - MLD protocol activity
  - <add-del-oif>[<all><clear>] <clear> <detail> <down-port> <error> <group> <level> <mcache-group> <mcache-source> <packet> <phy-port> <prime-port> <show> <source> <timer> <vlan>
- nd - neighbor discovery
- packet - IPv6 packet
- ra - router add

IPv6 Web Management using HTTP and HTTPS

Platform Support:

- FGS and FLS devices running software release 04.0.00 and later

When you have an IPv6 management station connected to a switch with an IPv6 address applied to the management port, you can manage the switch from a Web browser by entering **http://[**<ipv6 address>**]** or **https://[<ipv6 address>]** in the browser address field.

NOTE: You must enclose the IPv6 address with square brackets [ ] in order for the Web browser to work.

Restricting Web Access

Platform Support:

- FGS and FLS devices running software release 04.0.00 and later

You can restrict Web management access to include only management functions on a Foundry device that is acting as an IPv6 host, or restrict access so that the Foundry host can be reached by a specified IPv6 device.

Restricting Web Management Access by Specifying an IPv6 ACL

You can specify an IPv6 ACL that restricts Web management access to management functions on the device that is acting as the IPv6 host. For example:

```
FastIron(config)# access-list 12 deny host 2000:2383:e0bb::2/128 log
FastIron(config)# access-list 12 deny 30ff:3782::ff89/128 log
FastIron(config)# access-list 12 deny 3000:4828::fe19/128 log
FastIron(config)# access-list 12 permit any
FastIron(config)# web access-group ipv6 12
```

Syntax: web access-group ipv6 <ipv6 ACL name>

where <ipv6 ACL name> is a valid IPv6 ACL.

Restricting Web Management Access to an IPv6 Host

You can specify a single device with an IPv6 address to have Web management access to the host device. No other device except the one with the specified IPv6 address can access the Foundry device's Web management interface. For example:

```
FastIron(config)# web client ipv6 3000:2383:e0bb::2/128
```

Syntax: web client ipv6 <ipv6-address>

The <ipv6-address> you specify must be in hexadecimal format using 16-bit values between colons as documented in RFC 2373.

IPv6 Logging

Platform Support:

- FGS and FLS devices running software release 04.0.00 and later

This feature allows you to specify an IPv6 server as the Syslog server.

Specifying an IPv6 Syslog Server

To specify an IPv6 Syslog server, enter the log host ipv6 command as shown below:

EXAMPLES:

```
FastIron(config)#log host ipv6 2000:2383:e0bb::4/128
```

Syntax: [no] log host ipv6 <ipv6-address> [<udp-port-num>]

The <ipv6-address> must be in hexadecimal using 16-bit values between colons as documented in RFC 2373.

The <udp-port-num> optional parameter specifies the UDP application port used for the Syslog facility.

Possible values: See above.

Default value: N/A

Name-to-IPv6 Address Resolution using IPv6 DNS Server

The Domain Name Server (DNS) resolver feature lets you use a host name to perform Telnet, ping, and traceroute commands. You can also define a DNS domain on a Foundry device and thereby recognize all hosts within that domain. After you define a domain name, the Foundry device automatically appends the appropriate domain to the host and forwards it to the domain name server.

For example, if the domain “newyork.com” is defined on a Foundry device, and you want to initiate a ping to host “NYC01” on that domain, you need to reference only the host name in the command instead of the host name and its domain name. For example, you could enter either of the following commands to initiate the ping:

```
FastIron#ping nyc01
FastIron#ping nyc01.newyork.com
```

Defining an IPv6 DNS Entry

IPv6 defines new DNS record types to resolve queries for domain names to IPv6 addresses, as well as IPv6 addresses to domain names. Foundry devices running IPv6 software support AAAA DNS records, which are defined in RFC 1886.

AAAA DNS records are analogous to the A DNS records used with IPv4. They store a complete IPv6 address in each record. AAAA records have a type value of 28.

To establish an IPv6 DNS entry for the device, enter the following command:

```
FastIron(config)#ipv6 dns domain-name companynet.com
```

Syntax: [no] ipv6 dns domain-name <domain name>

To define an IPv6 DNS server address, enter the following command:

```
FastIron(config)#ipv6 dns server-address 200::1
```

Syntax: [no] ipv6 dns server-address <ipv6-addr> [<ipv6-addr>] [<ipv6-addr>] [<ipv6-addr>]

As an example, in a configuration where **ftp6.companynet.com** is a server with an IPv6 protocol stack, when a user pings ftp6.companynet.com, the Foundry device attempts to resolve the AAAA DNS record. In addition, if the DNS server does not have an IPv6 address, as long as it is able to resolve AAAA records, it can still respond to DNS queries.

IPv6 Ping

The **ping** command allows you to verify the connectivity from a Foundry device to an IPv6 device by performing an ICMP for IPv6 echo test.

For example, to ping a device with the IPv6 address of 2001:3424:847f:a385:34dd::45 from the Foundry device, enter the following command:

```
FastIron#ping ipv6 2001:3424:847f:a385:34dd::45
```

Syntax: ping ipv6 <ipv6-address> [outgoing-interface [<port> | ve <number>]] [source <ipv6-address>] [count <number>] [timeout <milliseconds>] [ttl <number>] [size <bytes>] [quiet] [numeric] [no-fragment] [verify] [data <1-to-4 byte hex>] [brief]

- The <ipv6-address> parameter specifies the address of the router. You must specify this address in hexadecimal using 16-bit values between colons as documented in RFC 2373.
- The **outgoing-interface** keyword specifies a physical interface over which you can verify connectivity. If you specify a physical interface, such as an Ethernet interface, you must also specify the port number of the interface. If you specify a virtual interface, such as a VE, you must specify the number associated with the VE.
- The **source** <ipv6-address> parameter specifies an IPv6 address to be used as the origin of the ping packets.
- The **count** <number> parameter specifies how many ping packets the router sends. You can specify from 1 - 4294967296. The default is 1.
- The **timeout** <milliseconds> parameter specifies how many milliseconds the router waits for a reply from the pinged device. You can specify a timeout from 1 - 4294967296 milliseconds. The default is 5000 (5 seconds).

- The **ttl** <number> parameter specifies the maximum number of hops. You can specify a TTL from 1 - 255. The default is 64.
- The **size** <bytes> parameter specifies the size of the ICMP data portion of the packet. This is the payload and does not include the header. You can specify from 0 - 4000. The default is 16.
- The **no-fragment** keyword turns on the "don't fragment" bit in the IPv6 header of the ping packet. This option is disabled by default.
- The **quiet** keyword hides informational messages such as a summary of the ping parameters sent to the device, and instead only displays messages indicating the success or failure of the ping. This option is disabled by default.
- The **verify** keyword verifies that the data in the echo packet (the reply packet) is the same as the data in the echo request (the ping). By default the device does not verify the data.
- The **data** <1 - 4 byte hex> parameter lets you specify a specific data pattern for the payload instead of the default data pattern, "abcd", in the packet's data payload. The pattern repeats itself throughout the ICMP message (payload) portion of the packet.

NOTE: For parameters that require a numeric value, the CLI does not check that the value you enter is within the allowed range. Instead, if you do exceed the range for a numeric value, the software rounds the value to the nearest valid value.

- The **brief** keyword causes ping test characters to be displayed. The following ping test characters are supported:

  ! Indicates that a reply was received.

  . Indicates that the network server timed out while waiting for a reply.

  U Indicates that a destination unreachable error PDU was received.

  I Indicates that the user interrupted ping.

SNMP3 over IPv6

Foundry FastIron devices support IPv6 for SNMP version 3. For more information about how to configure SNMP, see the chapter “Securing SNMP Access” on page 47-1.

Secure Shell and IPv6

Secure Shell (SSH) is a mechanism that allows secure remote access to management functions on the Foundry device. SSH provides a function similar to Telnet. You can log in to and configure the Foundry device using a publicly or commercially available SSH client program, just as you can with Telnet. However, unlike Telnet, which provides no security, SSH provides a secure, encrypted connection to the Foundry device.

To open an SSH session between an IPv6 host running an SSH client program and the Foundry device, open the SSH client program and specify the IPv6 address of the device. For more information about configuring SSH on the Foundry device, see “Configuring SSHv2 and SCP” on page 41-1.

IPv6 Telnet

Telnet sessions can be established between a Foundry device to a remote IPv6 host, and from a remote IPv6 host to the Foundry device using IPv6 addresses.

The **telnet** command establishes a Telnet connection from a Foundry device to a remote IPv6 host using the console. Up to five ***read-access*** Telnet sessions are supported on the router at one time. ***Write-access*** through Telnet is limited to one session, and only one outgoing Telnet session is supported on the router at one time. To see the number of open Telnet sessions at any time, enter the **show telnet** command.

EXAMPLES:

To establish a Telnet connection to a remote host with the IPv6 address of 3001:2837:3de2:c37::6, enter the following command:

```
FastIron#telnet 3001:2837:3de2:c37::6
```

Syntax: telnet <ipv6-address> [<port-number> | outgoing-interface ethernet <port> | ve <number>]

The <ipv6-address> parameter specifies the address of a remote host. You must specify this address in hexadecimal using 16-bit values between colons as documented in RFC 2373.

The <port-number> parameter specifies the port number on which the Foundry device establishes the Telnet connection. You can specify a value between 1 - 65535. If you do not specify a port number, the Foundry device establishes the Telnet connection on port 23.

If the IPv6 address you specify is a link-local address, you must specify the **outgoing-interface** ethernet <port> | ve <number> parameter. This parameter identifies the interface that must be used to reach the remote host. If you specify an Ethernet interface, you must also specify the port number associated with the interface. If you specify a VE interface, also specify the VE number.

Establishing a Telnet Session From an IPv6 Host

To establish a Telnet session from an IPv6 host to the Foundry device, open your Telnet application and specify the IPv6 address of the Layer 3 Switch.

IPv6 Traceroute

The **traceroute** command allows you to trace a path from the Foundry device to an IPv6 host.

The CLI displays trace route information for each hop as soon as the information is received. Traceroute requests display all responses to a minimum TTL of 1 second and a maximum TTL of 30 seconds. In addition, if there are multiple equal-cost routes to the destination, the Foundry device displays up to three responses.

For example, to trace the path from the Foundry device to a host with an IPv6 address of 3301:23dd:349e:a384::34, enter the following command:

```
FastIron#traceroute ipv6 3301:23dd:349e:a384::34
```

Syntax: traceroute ipv6 <ipv6-address>

The <ipv6-address> parameter specifies the address of a host. You must specify this address in hexadecimal using 16-bit values between colons as documented in RFC 2373.

IPv6 Management Commands

The following management CLI commands are available in FastIron devices that support IPv6:

- show ipv6 traffic
- clear ipv6 traffic
- show ipv6 TCP
- show ipv6 access-list
- show ipv6 mld-snooping
- clear ipv6 mld-snooping
- show ipv6 neighbor
- clear ipv6 neighbor

Chapter 7

Monitoring Hardware Components

The procedures in this chapter describe how to configure the software to monitor hardware components on FastIron devices. You can configure the software to do the following:

- Detect and report statistics about a cable connected to a copper port
- Monitor temperature and signal power levels for optical transceivers

Table 7.1 lists which FastIron devices support the features discussed in this chapter.

Table 7.1: Hardware Components Monitoring Support for FastIron Devices

Feature	FESX/ FSX/ FWSX	FGS and FLS
Virtual cable testing	X	<<?>>
Digital optical monitoring	X	<<?>>

Virtual Cable Testing

FastIron devices support ***Virtual Cable Test*** (VCT) technology. VCT technology enables the diagnosis of a conductor (wire or cable) by sending a pulsed signal into the conductor, then examining the reflection of that pulse. This method of cable analysis is referred to as Time Domain Reflectometry (TDR). By examining the reflection, the Foundry device can detect and report cable statistics such as local and remote link pair, cable length, and link status.

Configuration Notes

- This feature is supported on copper ports only. It is not supported on fiber ports.
- The port to which the cable is connected must be enabled when you issue the command to diagnose the cable. If the port is disabled, the command is rejected.
- If the port is operating at 100 Mbps half-duplex, the TDR test on one pair will fail.
- If the remote pair is set to forced 100 Mbps, any change in MDI/MDIX may cause the device to interpret the Multilevel Threshold-3 (MLT-3) as a reflected pulse, in which case, the device will report a faulty condition. In this scenario, it is recommended that you run the TDR test a few times for accurate results.

Command Syntax

To diagnose a cable using TDR, enter commands such as the following at the Privileged EXEC level of the CLI:

```
FastIron# phy cable-diag tdr 1
```

The above command diagnoses the cable attached to port 1.

When you issue the **phy-cable-diag** command, the command brings the port down for a second or two, then immediately brings the port back up.

Syntax: phy cable-diag tdr <port-num>

For <port-num>, specify the port in one of the following formats:

- FastIron GS and LS compact switches – <stacknum/slotnum/portnum>
- FastIron chassis devices – <slotnum/portnum>
- FESX, and FWSX compact switches – <portnum>

Viewing the Results of the Cable Analysis

To display the results of the cable analysis, enter a command such as the following at the Privileged EXEC level of the CLI:

```
FastIron#sh cable-diag tdr 1

Port      Speed Local pair Pair Length Remote pair Pair status
--------- ----- ---------- ----------- ----------- -----------
01        1000M Pair A     <50M        Pair B      Terminated
                Pair B     <50M        Pair A      Terminated
                Pair C     <50M        Pair D      Terminated
                Pair D     <50M        Pair C      Terminated
```

Syntax: show cable-diag tdr <port-num>

For <port-num>, specify the port in one of the following formats:

- FastIron GS and LS compact switches – <stacknum/slotnum/portnum>
- FastIron chassis devices – <slotnum/portnum>
- FESX, and FWSX compact switches – <portnum>

Table 7.2 defines the fields shown in the command output.

Table 7.2: Cable Statistics

This Line...	Displays...
Port	The port that was tested.
Speed	The port’s current line speed.
Local pair	The local link name.
Pair Length	The cable length when terminated, or the distance to the point of fault when the line is not up.
Remote pair	The remote link name.

Table 7.2: Cable Statistics (Continued)

This Line...	Displays...
Pair status	The status of the link. This field displays one of the following: • Terminated: The link is up. • Shorted: A short is detected in the cable. • Open: An opening is detected in the cable. • ImpedMis: The impedance is mismatched. • Failed: The TDR test failed.

Digital Optical Monitoring

You can configure your Foundry device to monitor optical transceivers in the system, either globally or by specified port(s). When this feature is enabled, the system will monitor the temperature and signal power levels for the optical transceivers in the specified port(s). Console messages and syslog messages are sent when optical operating conditions fall below or rise above the XFP or SFP manufacturer’s recommended thresholds.

Supported Media

Digital optical monitoring is supported with the following Foundry-qualified media types:

- 1000Base-BX-D
- 1000Base-BX-U
- 1000Base-LHA
- 1000Base-LX
- 1000Base-SX
- 1000Base-SX 2
- 100Base-FX-SR (support added in FSX 04.1.00)
- 100Base-FX-IR (support added in FSX 04.1.00)
- 10GBase-ER
- 10GBase-LR
- 10GBase-SR
- 10GBase-ZR
- 10GBase-ZRD

Configuration Limitations

A Foundry chassis device can monitor a maximum of 24 SFPs and 12 XFPs.

Enabling Digital Optical Monitoring

To enable optical monitoring on all Foundry-qualified optics installed in the device, use the following command:

```
FastIron(config)#optical-monitor
```

To enable optical monitoring on a specific port, use the following command:

```
FastIron(config)#interface ethernet 1/1
FastIron(config-if-e10000-1/1)#optical-monitor
```

To enable optical monitoring on a range of ports, use the following command:

```
FastIron(config)#interface ethernet 1/1 to 1/2
FastIron(config-mif-e10000-1/1-1/2)#optical-monitor
```

Syntax: [no] optical-monitor

Use the **no** form of the command to disable digital optical monitoring.

Setting the Alarm Interval

You can optionally change the interval between which alarms and warning messages are sent. The default interval is three minutes. To change the interval, use the following command:

```
FastIron(config)#interface ethernet 1/1 to 1/2
FastIron(config-mif-e10000-1/1-1/2)#optical-monitor 10
```

Syntax: [no] optical-monitor [<alarm-interval>]

For <alarm-interval>, enter a value between 1 and 65535. Enter 0 to disable alarms and warning messages.

NOTE: The commands **no optical-monitor** and **optical-monitor 0** perform the same function. That is, they both disable digital optical monitoring.

Displaying the Media Installed in a Port

You can use the **show media** command to obtain information about the media devices installed in a port. The following example output is from a FastIron X Series device running software release 04.1.00. The display output on your device may differ, depending on the software version running on the device.

```
FastIron#show media
Port   1: Type  : 1G M-SX2(SFP)
          Vendor:    Foundry Networks   Version: 0000
          Part# :    TRPAG1XRPBSS-FY    Serial#: 0635000468
Port   2: Type  : EMPTY
Port   3: Type  : EMPTY
Port   4: Type  : 100M M-FX-SR(SFP)
          Vendor:    Foundry Networks   Version: A
          Part# :    FTLF1217P2BTL-F1   Serial#: UCQ003A
Port   5: Type  : 1G M-C
Port   6: Type  : 1G M-C
Port   7: Type  : 1G M-C
Port   8: Type  : 1G M-C
Port   9: Type  : 1G M-C
Port  10: Type  : 1G M-C
Port  11: Type  : 1G M-C
Port  12: Type  : 1G M-C
Port  13: Type  : 1G M-C
Port  14: Type  : 1G M-C
Port  15: Type  : 1G M-C
Port  16: Type  : 1G M-C
Port  17: Type  : 1G M-C
Port  18: Type  : 1G M-C
Port  19: Type  : 1G M-C
Port  20: Type  : 1G M-C
Port  21: Type  : 1G M-C
Port  22: Type  : 1G M-C
Port  23: Type  : 1G M-C
Port  24: Type  : 1G M-C
Port  25: Type  : 10G XG-SR(XFP)
          Vendor:    Foundry Networks   Version: 02
          Part# :    JXPR01SW05306      Serial#: F617604000A3
Port  26: Type  : EMPTY
```

The results displayed from this command provide the Type, Vendor, Part number, Version and Serial number of the SFP or XFP optical device installed in the port. **1G M-C** indicates 1 Gigabit copper media. If no SFP or XFP device is installed in a port, the “Type” field will display “EMPTY”.

Viewing Optical Monitoring Information

To view temperature and power information for all qualified XFPs and SFPs in a particular slot, use the **show optic** command. The following shows an example output.

```
FastIron#show optic 4
 Port  Temperature   Tx Power     Rx Power     Tx Bias Current
+----+-----------+----------+------------+-------------------+
4/1   30.8242 C  -001.8822 dBm -002.5908 dBm   41.790 mA
       Normal       Normal         Normal         Normal
4/2   31.7070 C  -001.4116 dBm -006.4092 dBm   41.976 mA
       Normal       Normal         Normal         Normal
4/3   30.1835 C                -000.5794 dBm    0.000 mA
       Normal      Low-Alarm      Normal        Low-Alarm
4/4    0.0000 C                                 0.000 mA
       Normal       Normal         Normal         Normal
```

Syntax: show optic [<slot number>]

NOTE: This function takes advantage of information stored and supplied by the manufacturer of the XFP or SFP transceiver. This information is an optional feature of the Multi-Source Agreement standard defining the optical interface. Not all component suppliers have implemented this feature set. In such cases where the XFP or SFP transceiver does not supply the information, a "Not Available" message will be displayed for the specific port on which the module is installed.

The following table describes the information displayed by the **show optic** command.

Table 7.3: Output from the show optic command

This Field...	Displays...
Port	The Foundry port number.
Temperature	• The operating temperature, in degrees Celsius, of the optical transceiver. • The alarm status, as described in Table 7.4.
Tx Power	• The transmit power signal, in decibels (dB), of the measured power referenced to one milliwatt (mW). • The alarm status, as described in Table 7.4.
Rx Power	• The receive power signal, in decibels (dB), of the measured power referenced to one milliwatt (mW). • The alarm status, as described in Table 7.4
Tx Bias Current	• The transmit bias power signal, in milliamperes (mA). • The alarm status, as described in Table 7.4.

For Temperature, Tx Power, Rx Power, and Tx Bias Current in the **show optic** command output, values are displayed along with one of the following alarm status values: Low-Alarm, Low-Warn, Normal, High-Warn or High-

Alarm. The thresholds that determine these status values are set by the manufacturer of the optical transceivers. Table 7.4 describes each of these status values.

Table 7.4: Alarm Status Value Description

Status Value	Description
Low-Alarm	Monitored level has dropped below the "low-alarm" threshold set by the manufacturer of the optical transceiver.
Low-Warn	Monitored level has dropped below the "low-warn" threshold set by the manufacturer of the optical transceiver.
Normal	Monitored level is within the "normal" range set by the manufacturer of the optical transceiver.
High-Warn	Monitored level has climbed above the "high-warn" threshold set by the manufacturer of the optical transceiver.
High-Alarm	Monitored level has climbed above the "high-alarm" threshold set by the manufacturer of the optical transceiver.

Viewing Optical Transceiver Thresholds

The thresholds that determine the alarm status values for an optical transceiver are set by the manufacturer of the XFP or SFP. To view the thresholds for a qualified optical transceiver in a particular port, use the **show optic threshold** command as shown below.

```
FastIron#show optic threshold 2/2
Port 2/2 sfp monitor thresholds:
Temperature High alarm                5a00          90.0000 C
Temperature Low alarm                 d300         -45.0000 C
Temperature High warning              5500          85.0000 C
Temperature Low warning               d800         -40.0000 C
Supply Voltage High alarm             9088
Supply Voltage Low alarm              7148
Supply Voltage High warning           8ca0
Supply Voltage Low warning            7530
TX Bias High alarm                    7530          60.000 mA
TX Bias Low alarm                     01f4           1.000 mA
TX Bias High warning                  61a8          50.000 mA
TX Bias Low warning                   05dc           3.000 mA
TX Power High alarm                   1f07        -001.0001 dBm
TX Power Low alarm                    02c4        -011.4996 dBm
TX Power High warning                 18a6        -001.9997 dBm
TX Power Low warning                  037b        -010.5012 dBm
RX Power High alarm                   2710         000.0000 dBm
RX Power Low alarm                    0028        -023.9794 dBm
RX Power High warning                 1f07        -001.0001 dBm
RX Power Low warning                     0032          -023.0102 dBm
```

Syntax: show optic threshold <port-num>

For <port-num>, specify the port in one of the following formats:

- FastIron GS and LS compact switches – <stacknum/slotnum/portnum>
- FastIron chassis devices – <slotnum/portnum>
- FESX, and FWSX compact switches – <portnum>

For Temperature, Supply Voltage, TX Bias, TX Power, and RX Power, values are displayed for each of the following four alarm and warning settings: High alarm, Low alarm, High warning, and Low warning. The hexadecimal values are the manufacturer's internal calibrations, as defined in the SFF-8472 standard. The other values indicate at what level (above the high setting or below the low setting) the system should send a warning message or an alarm. Note that these values are set by the manufacturer of the optical transceiver, and cannot be configured.

Syslog Messages

The system generates Syslog messages for optical transceivers when:

- The temperature, supply voltage, TX Bias, TX power, or TX power value goes above or below the high or low warning or alarm threshold set by the manufacturer.
- The optical transceiver does not support digital optical monitoring.
- The optical transceiver is not qualified, and therefore not supported by Foundry.

For details about the above Syslog messages, see the chapter *"Using Syslog" on page A-1*.

Chapter 8

Configuring Basic Layer 2 Features

The procedures in this chapter describe how to configure basic Layer 2 parameters.

Foundry devices are configured at the factory with default parameters that allow you to begin using the basic features of the system immediately. However, many of the advanced features such as VLANs or routing protocols for the device must first be enabled at the system (global) level before they can be configured. If you use the Command Line Interface (CLI) to configure system parameters, you can find these system level parameters at the Global CONFIG level of the CLI.

NOTE:

- Before assigning or modifying any router parameters, you must assign the IP subnet (interface) addresses for each port.
- For information about configuring IP addresses, DNS resolver, DHCP assist, and other IP-related parameters, see the chapter “Configuring IP” on page 29-1.
- For information about the Syslog buffer and messages, see “Using Syslog” on page A-1.

About Port Regions

Ports on the X Series devices are grouped into regions. For a few features, you will need to know the region to which a port belongs. However, for most features, a port’s region does not affect configuration or operation of the feature.

NOTE: Port regions do not apply to trunk group configurations on the X Series devices. However, port regions do apply to port monitoring and unknown unicast configurations.

FastIron Edge Switch X424 and X424HF, and FastIron Workgroup Switch X424:

- Ports 1 – 12
- Ports 13 – 24
- Port 25
- Port 26

FastIron Edge Switch X448 and FastIron Workgroup Switch X448:

- Ports 1 – 12

- Ports 13 – 24
- Port 25 – 36
- Port 37 – 48
- Port 49
- Port 50

FastIron SuperX:

- Management Module:
    - Ports 1 – 12
- 24-port Gigabit Ethernet Copper Interface Module
    - Ports 1 – 12
    - Ports 13 – 24
- 24-port Gigabit Ethernet Fiber Interface Module:
    - Ports 1 – 12
    - Ports 13 – 24
- 2-port 10-Gigabit Ethernet Fiber Interface Module
    - Port 1
    - Port 2

Enabling or Disabling the Spanning Tree Protocol (STP)

STP (IEEE 802.1D bridge protocol) is supported on all Foundry devices. STP detects and eliminates logical loops in the network. STP also ensures that the least cost path is taken when multiple paths exist between ports or VLANs. If the selected path fails, STP searches for and then establishes an alternate path to prevent or limit retransmission of data.

NOTE: This section provides instructions for enabling and disabling STP. For configuration procedures and information about Foundry's STP, see the chapter "Configuring Spanning Tree Protocol (STP) Related Features" on page 9-1 in this guide.

STP must be enabled at the system level to allow assignment of this capability on the VLAN level. On devices running Layer 2 code, STP is enabled by default. On devices running Layer 3 code, STP is disabled by default.

To enable STP for all ports on a Foundry device:

```
FastIron(config)#spanning tree
```

Syntax: [no] spanning-tree

You can also enable and disable spanning tree on a port-based VLAN and on an individual port basis, and enable advanced STP features. See "Configuring Spanning Tree Protocol (STP) Related Features" on page 9-1.

Modifying STP Bridge and Port Parameters

You can modify the following STP Parameters:

- Bridge parameters – forward delay, maximum age, hello time, and priority
- Port parameters – priority and path cost

For configuration details, see "Changing STP Bridge and Port Parameters" on page 9-4.

MAC Learning Rate Control

FGS software release 03.0.00 adds support for a new CLI command that allows users to set a rate limit to control CPU address updating. The range for this rate limit is 200 - 50,000 per second. The MAC learning rate limit applies to each packet processor, which means that for a system with two packet processors, each processor can send address messages to the CPU at the established rate limit.

Syntax: [no] cpu-limit addr-msgs <msgsRateLimit>

NOTE: Actual rates in hardware may have a variance of +200 or -100.

Changing the MAC Age Time and Disabling MAC Address Learning

You can change the MAC address age timer using the **mac-age-time** command.

- On FastIron X Series devices running pre-release 02.5.00 software, and on the FastIron GS, learned MAC address entries do not age out until they are unused for 300 – 600 seconds.
- On FastIron X Series devices running software release 02.5.00 or later, the maximum number of seconds that can be allocated to the MAC age timer has increased from 600 seconds to 14000 seconds. In addition, you can specify the MAC age timer in 10 second intervals, whereas releases prior to 02.5.00 allow 60 second intervals only. For example, in release 02.5.00, you can specify 10, 20, or 14000 as the MAC age timer, but not 122. In releases prior to 02.5.00, you can specify 60 and 120, but not 100.

To change the MAC age time, enter a command such as the following:

```
FastIron(config)#mac-age-time 60
```

Syntax: [no] mac-age-time <secs>

<secs> specifies the number of seconds. Possible values differ depending on the version of software running on your device, as follows:

- On FastIron X Series devices running pre-release 02.5.00 software, and on FastIron GS devices, you can configure 0 or a value from 60 – 600 (seconds), in 60-second intervals. If you set the MAC age time to 0, aging is disabled.
- On FastIron X Series devices running software release 02.5.00 through 03.1.00*x*, you can configure 0 or a value from 10 – 14000 (seconds), in 10-second intervals. If you set the MAC age time to 0, aging is disabled.
- On FastIron X Series devices running software FSX release 03.2.00 or later, you can configure 0 or a value from 10 – 86,400 (seconds), in 10-second intervals. If you set the MAC age time to 0, aging is disabled.

NOTE: Usually, the actual MAC age time is from one to two times the configured value. For example, if you set the MAC age timer to 60 seconds, learned MAC entries age out after remaining unused for between 60 – 120 seconds. However, if all of the following conditions are met, then the MAC entries age out after a longer than expected duration:

- The MAC age timer is greater than 630 seconds.
- The number of MAC entries is over 6000.
- All MAC entries are learned from the same packet processor.
- All MAC entries age out at the same time.

Disabling the Automatic Learning of MAC Addresses

Platform Support:

- FESX/FSX/FWSX devices running software release 03.2.00 and later
- FGS and FLS devices running software release 04.0.00 and later

By default, when a packet with an unknown Source MAC address is received on a port, the Foundry device learns this MAC address on the port.

You can prevent a physical port from learning MAC addresses by entering the following command:

```
FastIron(config)#interface ethernet 3/1
FastIron(config-if-e1000-3/1)#mac-learn-disable
```

Syntax: [no] mac-learn disable

Use the no form of the command to allow a physical port to learn MAC addresses.

Configuration Notes and Feature Limitations

- This command is not available on virtual routing interfaces. Also, if this command is configured on the primary port of a trunk, MAC address learning will be disabled on all the ports in the trunk.
- Entering the **mac-learn-disable** command on tagged ports disables MAC learning for that port in all VLANs to which that port is a member. For example, if tagged port 3/1 is a member of VLAN 10, 20, and 30 and you issue the **mac-learn-disable** command on port 3/1, port 3/1 will not learn MAC addresses, even if it is a member of VLAN 10, 20, and 30.

Displaying the MAC Address Table

To display the MAC table, enter the following command:

```
FastIron#show mac-address
Total active entries from all ports = 3
Total static entries from all ports = 1
  MAC-Address     Port      Type   VLAN
1234.1234.1234     15   Static      1
0004.8038.2f24     14  Dynamic      1
0004.8038.2f00     13  Dynamic      1
0010.5a86.b159     10  Dynamic      1
```

In the output of the **show mac-address** command, the *Type* column indicates whether the MAC entry is static or dynamic. A static entry is one you create using the **static-mac-address** command. A dynamic entry is one that is learned by the software from network traffic.

The output of the **show mac-address** command on FESX, FSX, and FWSX devices include an *Index* column which indicates the index where the entry exists in the hardware MAC table.

NOTE: The **show mac-address** command output does not include MAC addresses for management ports, since these ports do not support typical MAC learning and MAC-based forwarding.

Configuring Static MAC Entries

Static MAC addresses can be assigned to Foundry devices.

NOTE: Foundry devices running Layer 3 code also support the assignment of static IP Routes, static ARP, and static RARP entries. For details on configuring these types of static entries, see “Configuring Static Routes” on page 29-35 and “Creating Static ARP Entries” on page 29-29.

You can manually input the MAC address of a device to prevent it from being aged out of the system address table.

This option can be used to prevent traffic for a specific device, such as a server, from flooding the network with traffic when it is down. Additionally, the static MAC address entry is used to assign higher priorities to specific MAC addresses.

You can specify traffic priority (QoS) and VLAN membership (VLAN ID) for the MAC Address as well as specify the device type of either router or host.

The default and maximum configurable MAC table sizes can differ depending on the device. To determine the default and maximum MAC table sizes for your device, display the system parameter values. See "Displaying and Modifying System Parameter Default Settings" on page 8-11.

Static Multicast MAC

Platform Support:

- FGS/FLS devices running software release 04.1.00 and later
- FESX/FSX/FWSX devices running software release 04.1.00 and later – L2, BL3, L3

Many applications, such as Microsoft NLB, Juniper IPS, and Netscreen Firewall, use the same MAC address to announce load-balancing services. As a result, a switch must be able to learn the same MAC address on several ports. Static Multicast MAC allows you to statically configure a MAC address on multiple ports using a single command.

Configuration Notes

- The FastIron GS and LS support a maximum of 7 static multicast MAC addresses.
- FastIron X Series devices support a maximum of 15 static multicast MAC addresses.
- Hosts or physical interfaces normally join multicast groups dynamically, but you can also statically configure a host or an interface to join a multicast group.

Command Syntax

For example, to add a static entry for a server with a MAC address of 0045.5563.67ff and a priority of 7, enter the following command:

```
FastIron(config)#static-mac-address 0045.5563.67ff ethernet 4/2 ethernet 4/3
ethernet 4/4 priority 7
```

Syntax: [no] static-mac-address <mac-addr> ethernet [<slotnum>/]<portnum> ethernet [<slotnum>/]<portnum> ethernet [<slotnum>/]<portnum> …. priority <num>

The <slotnum> parameter is required on chassis devices.

The <portnum> parameter is a valid port number.

The priority <num> can be 0 – 7 (0 is lowest priority and 7 is highest priority). The default priority is 0.

NOTE: The location of the **static-mac-address** command in the CLI depends on whether you configure port-based VLANs on the device. If the device does not have more than one port-based VLAN (VLAN 1, which is the default VLAN that contains all the ports), the **static-mac-address** command is at the global CONFIG level of the CLI. If the device has more than one port-based VLAN, then the **static-mac-address** command is not available at the global CONFIG level. In this case, the command is available at the configuration level for each port-based VLAN.

Configuring VLAN-based Static MAC Entries

Platform Support:

- FESX/FSX/FWSX devices running software release 04.0.00 and later – L2, BL3, and L3
- FGS and FLS devices running software release 04.0.00 and later

You can configure a VLAN to drop packets that have a particular source or destination MAC address.

You can configure a maximum of 2048 static MAC address drop entries on a Foundry device.

Use the CLI command **show running-config** to view the static MAC address drop entries currently configured on the device.

Command Syntax

To configure a VLAN to drop packets with a source or destination MAC address of 1145.5563.67FF, enter the following commands:

```
FastIron(config)#vlan 2
FastIron(config-vlan-2)#static-mac-address 1145.5563.67FF drop
```

Syntax: [no] static-mac-address <mac-addr> drop

Use the **no** form of the command to remove the static MAC address drop configuration.

Clearing MAC Address Entries

You can remove learned MAC address entries from the MAC address table. The types of MAC address can be removed are as follows:

- All MAC address entries
- All MAC address entries for a specified Ethernet port
- All MAC address entries for a specified VLAN
- All specified MAC address entry in all VLANs

For example, to remove entries for the MAC address 000d.cd80.00d0 in all VLANs, enter the following command at the Privilege EXEC level of the CLI:

```
FastIron#clear mac-address 000d.cb80.00d0
```

Syntax: clear mac-address <mac-address> | ethernet <port-num> | vlan <vlan-num>

If you enter **clear mac-address** without any parameter, the software removes all MAC address entries.

Use the <mac-address> parameter to remove a specific MAC address from all VLANs. Specify the MAC address in the following format: HHHH.HHHH.HHHH.

Use the **ethernet** <port-num> parameter to remove all MAC addresses for a specific Ethernet port.

Use the vlan <num> parameter to remove all MAC addresses for a specific VLAN.

Enabling Port-Based VLANs

When using the CLI, port and protocol-based VLANs are created by entering one of the following commands at the global CONFIG level of the CLI.

To create a port-based VLAN, enter commands such as the following:

```
FastIron(config)#vlan 222 by port
FastIron(config)#vlan 222 name Mktg
```

Syntax: vlan <num> by port

Syntax: vlan <num> name <string>

The <num> parameter specifies the VLAN ID. The valid range for VLAN IDs starts at 1 on all systems but the upper limit of the range differs depending on the device. In addition, you can change the upper limit on some devices using the **system max-vlans...** command.

The <string> parameter is the VLAN name and can be a string up to 32 characters. You can use blank spaces in the name if you enclose the name in double quotes (for example, “Product Marketing”.)

You can configure up to 4063 port-based VLANs on a device running Layer 2 code or 4061 port-based VLANs on a device running Layer 3 code. Each port-based VLAN can contain either tagged or untagged ports. A port cannot be a member of more than one port-based VLAN unless the port is tagged. On both device types, valid VLAN IDs are 1 – 4095. You can configure up to the maximum number of VLANs within that ID range.

NOTE: VLAN IDs 4087, 4090, and 4093 are reserved for Foundry internal use only. VLAN 4094 is reserved for use by Single STP. Also, in releases prior to 04.0.00, VLAN IDs 4091 and 4092 are reserved for Foundry internal use only. Starting in release 04.0.00, if you want to use VLANs 4091 and 4092 as configurable VLANs, you can assign them to different VLAN IDs. For more information, see "Assigning Different VLAN IDs to Reserved VLANs 4091 and 4092" on page 14-15.

NOTE: The second command is optional and also creates the VLAN if the VLAN does not already exist. You can enter the first command after you enter the second command if you first exit to the global CONFIG level of the CLI.

Assigning IEEE 802.1Q Tagging to a Port

When a port is tagged, it allows communication among the different VLANs to which it is assigned. A common use for this might be to place an email server that multiple groups may need access to on a tagged port, which in turn, is resident in all VLANs that need access to the server.

NOTE: Tagging does not apply to the default VLAN.

When using the CLI, ports are defined as either tagged or untagged at the VLAN level.

Command Syntax

Suppose you want to make port 5 a member of port-based VLAN 4, a tagged port. To do so, enter the following:

```
FastIron(config)#vlan 4
FastIron(config-vlan-4)#tagged e 5
```

Syntax: tagged ethernet [<slotnum>/]<portnum> [to [<slotnum>/]<portnum> [ethernet [<slotnum>/]<portnum>...]]

The <slotnum> parameter is required on chassis devices.

Defining MAC Address Filters

MAC layer filtering enables you to build access lists based on MAC layer headers in the Ethernet/IEEE 802.3 frame. You can filter on the source and destination MAC addresses. The filters apply to incoming traffic only.

You configure MAC filters globally, then apply them to individual interfaces. To apply MAC filters to an interface, you add the filters to that interface's MAC filter group.

The device takes the action associated with the first matching filter. If the packet does not match any of the filters in the access list, the default action is to drop the packet. If you want the system to permit traffic by default, you must specifically indicate this by making the last entry in the access list a permit filter. Here is an example:
mac filter <last-index-number> **permit any any**

For devices running Layer 3 code, the MAC filter is applied only to those inbound packets that are to be switched. This includes those ports associated with a virtual routing interface. However, the filter is not applied to the virtual routing interface. It is applied to the physical port.

When you create a MAC filter, it takes effect immediately. You do not need to reset the system. However, you do need to save the configuration to flash memory to retain the filters across system resets.

Configuration Notes and Limitations

- MAC filtering on FastIron devices is performed in hardware.
- Layer 2 MAC filtering on FastIron devices differ from other Foundry devices in that you can only filter on source and destination MAC addresses. Other Foundry devices allow you to also filter on the encapsulation

type and frame type.

- Layer 2 MAC filtering on FastIron devices differs from the FES and BigIron in that MAC filtering applies to all traffic, including management traffic. To exclude management traffic from being filtered, configure a MAC filter that explicitly permits all traffic headed to the management MAC (destination) address. The MAC address for management traffic is always the MAC address of port 1.

The following configuration notes apply to Foundry Layer 3 devices:

- Use MAC Layer 2 filters only for switched traffic. If a routing protocol (for example, IP) is configured on an interface, a MAC filter defined on that interface is not applied to inbound packets. If you want to filter inbound route traffic, configure a route filter.
- You cannot use Layer 2 filters to filter Layer 4 information.
- MAC Layer 2 filters are not supported on tagged ports in the base Layer 3 and full Layer 3 images.

Command Syntax

To configure and apply a MAC filter, enter commands such as the following:

```
FastIron(config)#mac filter 1 deny 3565.3475.3676 ffff.0000.0000
FastIron(config)#mac filter 1024 permit any any
FastIron(config)#int e 1
FastIron(config-if-e1000-1)#mac filter-group 1
```

These commands configure a filter to deny ARP traffic with a source MAC address that begins with “3565” to any destination. The second filter permits all traffic that is not denied by another filter.

NOTE: Once you apply a MAC filter to a port, the device drops all Layer 2 traffic on the port that does not match a MAC permit filter on the port.

Syntax: [no] mac filter <filter-num> permit | deny <src-mac> <mask> | any <dest-mac> <mask | any

The **permit | deny** argument determines the action the software takes when a match occurs.

The <src-mac> <mask> | **any** parameter specifies the source MAC address. You can enter a specific address value and a comparison mask or the keyword **any** to filter on all MAC addresses. Specify the mask using f’s (ones) and zeros. For example, to match on the first two bytes of the address aabb.ccdd.eeff, use the mask ffff.0000.0000. In this case, the filter matches on all MAC addresses that contain "aabb" as the first two bytes. The filter accepts any value for the remaining bytes of the MAC address. If you specify **any**, do not specify a mask. In this case, the filter matches on all MAC addresses.

The <dest-mac> <mask> | **any** parameter specifies the destination MAC address. The syntax rules are the same as those for the <src-mac> <mask> | **any** parameter.

Syntax: [no] mac filter log-enable

Globally enables logging for filtered packets.

Syntax: [no] mac filter-group log-enable

Enables logging for filtered packets on a specific port.

Syntax: [no] mac filter-group <filter-list>

Applies MAC filters to a port.

NOTE: The filters must be applied as a group. For example, if you want to apply four filters to an interface, they must all appear on the same command line.

NOTE: You cannot add or remove individual filters in the group. To add or remove a filter on an interface, apply the filter group again containing all the filters you want to apply to the port.

NOTE: If you apply a filter group to a port that already has a filter group applied, the older filter group is replaced by the new filter group.

When a MAC filter is applied to or removed from an interface, a Syslog message such as the following is generated:

```
SYSLOG: <14>Jan 1 00:00:00 10.44.9.11 MAC Filter applied to port 0/1/2 by tester
from telnet session (filter id=5 ).
```

```
SYSLOG: <14>Jan 1 00:00:00 10.44.9.11 MAC Filter removed from port 0/1/2 by tester
from telnet session (filter id=5 ).
```

The Syslog messages indicate that a MAC filter was applied to the specified port by the specified user during the specified session type. Session type can be Console, Telnet, SSH, Web, SNMP, or others. The filter IDs that were added or removed are listed.

Enabling Logging of Management Traffic Permitted by MAC Filters

You can configure the Foundry device to generate Syslog entries and SNMP traps for management traffic that is permitted by Layer 2 MAC filters. ***Management traffic*** applies to packets that are destined for the CPU, such as control packets. You can enable logging of permitted management traffic on a global basis or an individual port basis.

The first time an entry in a MAC filter permits a management packet and logging is enabled for that entry, the software generates a Syslog message and an SNMP trap. Messages for management packets permitted by MAC filters are at the warning level of the Syslog.

When the first Syslog entry for a management packet permitted by a MAC filter is generated, the software starts a five-minute timer. After this, the software sends Syslog messages every five minutes. The messages list the number of management packets permitted by each MAC filter during the previous five-minute interval. If a MAC filter does not permit any packets during the five-minute interval, the software does not generate a Syslog entry for that MAC filter.

NOTE: For a MAC filter to be eligible to generate a Syslog entry for permitted management packets, logging must be enabled for the filter. The Syslog contains entries only for the MAC filters that permit packets and have logging enabled.

When the software places the first entry in the log, the software also starts the five-minute timer for subsequent log entries. Thus, five minutes after the first log entry, the software generates another log entry and SNMP trap for permitted management packets.

Configuration Notes

MAC filter logging is supported in the following FastIron configurations:

- FESX devices running software release 02.1.01 or later
- All FSX devices and associated software releases
- All FWSX devices and associated software releases

These releases support MAC filter logging of management traffic only.

Command Syntax

To configure Layer 2 MAC filter logging globally, enter the following CLI commands at the global CONFIG level:

```
FastIron(config)#mac filter log-enable
FastIron(config)#write memory
```

Syntax: [no] mac filter log-enable

To configure Layer 2 MAC filter logging for MAC filters applied to ports 1 and 3, enter the following CLI commands:

```
FastIron(config)#int ethernet 1
FastIron(config-if-e1000-1)#mac filter-group log-enable
FastIron(config-if-e1000-1)#int ethernet 3
FastIron(config-if-e1000-3)#mac filter-group log-enable
FastIron(config-if-e1000-3)#write memory
```

Syntax: [no] mac filter-group log-enable

MAC Filter Override for 802.1X-Enabled Ports

Platform Support:

- FGS and FLS devices running software version 4.1 and later

The MAC filtering feature on an 802.1X-enabled port allows 802.1X and non-802.1X devices to share the same physical port. For example, this feature enables you to connect a PC and a non-802.1X device, such as a Voice Over IP (VOIP) phone, to the same 802.1X-enabled port on the Foundry device. The IP phone will bypass 802.1X authentication and the PC will require 802.1X authentication.

To enable this feature, first create a MAC filter, then bind it to an interface on which 802.1X is enabled. The MAC filter includes a mask that can match on any number of bytes in the MAC address. The mask can eliminate the need to enter MAC addresses for all non-802.1X devices connected to the Foundry device, and the ports to which these devices are connected.

Configuration Notes

- This feature is supported on untagged, tagged, and dual-mode ports.
- You can configure this feature on ports that have ACLs and MAC filters defined.

Configuration Syntax

To configure MAC filtering on an 802.1X-enabled port, enter commands such as the following:

```
FastIron#(config)#mac filter 1 permit 0050.04ab.9429 ffff.ffff.0000 any
FastIron#(config)#int e1/2
FastIron#(config-if-e1000-1/2)#dot1x auth-filter 1 3 to 5 10
```

The first line defines a MAC filter that matches on the first four bytes (ffff.ffff.0000) of the source MAC address 0050.04ab.9429, and any destination MAC address. The permit action creates an 802.1X session in the FORCE AUTHORIZE state, meaning that the device is placed unconditionally in the authorized state, bypassing 802.1X authentication and allowing all traffic from the specified MAC address. If no match is found, the implicit action is to authenticate the client.

The last line binds MAC filters 1, 3, 4, 5, and 10 to interface 2.

Syntax: **mac filter** <filter-num> **permit | deny** <src-mac> <mask> | **any** <dest-mac> <mask | **any**

Syntax: **dot1x auth-filter** <filter-list>

The **permit | deny** argument determines the action the software takes when a match occurs. In the previous example, the *permit* action creates an 802.1X session in the FORCE AUTHORIZE state, meaning that the device is placed unconditionally in the authorized state, bypassing 802.1X authentication and allowing all traffic from the specified MAC address. The *deny* action creates an 802.1X session in the FORCE UNAUTHORIZE state, meaning that the device will never be authorized, even if it has the appropriate credentials.

The <src-mac> <mask> | **any** parameter specifies the source MAC address. You can enter a specific address value and a comparison mask, or the keyword **any** to filter on all MAC addresses. Specify the mask using f (ones) and zeros. For example, to match on the first two bytes of the address aabb.ccdd.eeff, use the mask ffff.0000.0000. The filter matches on all MAC addresses that contain aabb as the first two bytes and accepts any value for the remaining bytes of the MAC address. If you specify **any**, do not specify a mask. In this case, the filter matches on all MAC addresses. If no match is found, the implicit action is to authenticate the client.

The <dest-mac> <mask> | **any** parameter specifies the destination MAC address. The syntax rules are the same as those for the <src-mac> <mask> | **any** parameter. Note that the 802.1x Authentication filter (**dot1x auth-filter**) does not use the destination MAC address in the MAC filter.

The <filter-num> command identifies the MAC filter. The maximum number of supported MAC filters is determined by the **mac-filter-sys** default or configured value.

The **dot1x auth-filter** <filter-list> command binds MAC filters to a port.

The following rules apply when using the **dot1x auth-filter** command:

- When you add filters to or modify the **dot1x auth-filter**, the system clears all 802.1X sessions on the port. Consequently, all users that are logged in will need to be re-authenticated.
- The maximum number of filters that can be bound to a port is limited by the **mac-filter-port** default or configured value.
- The filters must be applied as a group. For example, if you want to apply four filters to an interface, they must all appear on the same command line.
- You cannot add or remove individual filters in the group. To add or remove a filter on an interface, apply the filter group again containing all the filters you want to apply to the port.

If you apply a filter group to a port that already has a filter group applied, the older filter group is replaced by the new filter group.

Locking a Port to Restrict Addresses

Address-lock filters allow you to limit the number of devices that have access to a specific port. Access violations are reported as SNMP traps. This feature is disabled by default. A maximum of 2048 entries can be specified for access. The default address count is eight.

Configuration Notes

- Static trunk ports and link-aggregation configured ports on FastIron devices do not support the lock-address option.
- The MAC port security feature is a more robust version of this feature. See the chapter “Using the MAC Port Security Feature” on page 43-1.

Command Syntax

To enable address locking for port 2 and place a limit of 15 entries, enter a command such as the following:

```
FastIron(config)#lock e 2 addr 15
```

Syntax: lock-address ethernet [<stacknum>/<slotnum>/]<portnum> [addr-count <num>]

The <stacknum> parameter is required on FastIron GS devices running software release 02.5.00 or later.

The <slotnum> parameter is required on chassis devices and on FastIron GS devices running software release 02.5.00 or later.

The <portnum> parameter is a valid port number.

The <num> parameter is a value from 1 – 2048.

Displaying and Modifying System Parameter Default Settings

Foundry devices have default table sizes for the system parameters shown in the following display outputs. The table sizes determine the maximum number of entries the tables can hold. You can adjust individual table sizes to accommodate your configuration needs.

The tables you can configure, as well as the default values and valid ranges for each table, differ depending on the Foundry device you are configuring. To display the adjustable tables on your Foundry device, use the **show default values** command. The following shows example outputs.

Configuration Considerations

- Changing the table size for a parameter reconfigures the device's memory. Whenever you reconfigure the memory on a Foundry device, you must save the change to the startup-config file, then reload the software to place the change into effect.
- Configurable tables and their defaults and maximum values differ on FastIron IPv4 devices versus IPv6 devices.
- For more information about Layer 3 system parameter limits, see "Modifying and Displaying Layer 3 System Parameter Limits" on page 20-2.

Displaying System Parameter Default Values

To display the configurable tables and their defaults and maximum values, enter the **show default values** command at any level of the CLI.

The following shows an example output of the **show default values** command on a FastIron Layer 2 device.

```
FastIron#show default values
sys log buffers:50         mac age time:300 sec       telnet sessions:5

System Parameters    Default    Maximum    Current
igmp-max-group-addr  255        1024       255
l3-vlan              32         1024       32
mac                  16000      16000      16000
vlan                 64         4095       4095
spanning-tree        32         255        32
mac-filter-port      32         256        32
mac-filter-sys       64         512        64
view                 10         65535      10
```

The following shows an example output on a FastIron IPV4 device running Layer 3 software.

```
FastIron#show default values
sys log buffers:50          mac age time:300 sec        telnet sessions:5

ip arp age:10 min           bootp relay max hops:4      ip ttl:64 hops
ip addr per intf:24

when multicast enabled :
igmp group memb.:140 sec    igmp query:60 sec

when ospf enabled :
ospf dead:40 sec            ospf hello:10 sec           ospf retrans:5 sec
ospf transit delay:1 sec

when bgp enabled :
bgp local pref.:100         bgp keep alive:60 sec       bgp hold:180 sec
bgp metric:10               bgp local as:1              bgp cluster id:0
bgp ext. distance:20        bgp int. distance:200       bgp local distance:200

System Parameters     Default    Maximum    Current
ip-arp                4000       64000      4000
ip-static-arp         512        1024       512
atalk-route           1024       1536       1024
atalk-zone-port       64         255        64
atalk-zone-sys        768        2048       768
multicast-route       64         8192       64
dvmrp-route           2048       32000      2048
dvmrp-mcache          512        4096       512
pim-mcache            1024       4096       1024
ip-cache              80000      400000     80000
ip-filter-port        1015       1015       1015
ip-filter-sys         2048       8192       2048
ipx-forward-filter    32         128        32
ipx-rip-entry         2048       8192       2048
ipx-rip-filter        32         128        32
ipx-sap-entry         4096       8192       4096
ipx-sap-filter        32         128        32
l3-vlan               32         1024       32
ip-qos-session        1024       16000      1024
mac                   16384      16384      16384
ip-route              80000      262144     80000
ip-static-route       64         1024       64
vlan                  64         4095       64
spanning-tree         32         255        32
mac-filter-port       16         256        16
mac-filter-sys        32         512        32
ip-subnet-port        24         128        24
session-limit         65536      160000     65536
view                  10         65535      10
virtual-interface     255        512        255
hw-ip-next-hop        2048       6144       2048
hw-logical-interface  4096       4096       4096
hw-ip-mcast-mll       1024       4096       1024
hw-traffic-condition  50         1024       50
rmon-entries          2048       32768      2048
```

The following shows an example output on a FastIron IPV6 device running Layer 3 software. The lines in bold type indicate system parameter values that are different for IPv6 devices versus IPv4 devices.

```
FastIron#show default values
sys log buffers:50         mac age time:300 sec       telnet sessions:5

ip arp age:10 min          bootp relay max hops:4     ip ttl:64 hops
ip addr per intf:24

when multicast enabled :
igmp group memb.:140 sec   igmp query:60 sec

when ospf enabled :
ospf dead:40 sec           ospf hello:10 sec          ospf retrans:5 sec
ospf transit delay:1 sec

when bgp enabled :
bgp local pref.:100        bgp keep alive:60 sec      bgp hold:180 sec
bgp metric:10              bgp local as:1             bgp cluster id:0
bgp ext. distance:20       bgp int. distance:200      bgp local distance:200

System Parameters    Default    Maximum    Current
ip-arp               4000       64000      4000
ip-static-arp        512        1024       512
atalk-route          1024       1536       1024
atalk-zone-port      64         255        64
atalk-zone-sys       768        2048       768
multicast-route      64         8192       64
dvmrp-route          2048       32000      2048
dvmrp-mcache         512        4096       512
pim-mcache           1024       4096       1024
ip-cache             10000      256000     10000
ip-filter-port       1015       1015       1015
ip-filter-sys        2048       4096       2048
ipx-forward-filter   32         128        32
ipx-rip-entry        2048       8192       2048
ipx-rip-filter       32         128        32
ipx-sap-entry        4096       8192       4096
ipx-sap-filter       32         128        32
l3-vlan              32         1024       32
ip-qos-session       1024       16000      1024
mac                  16384      16384      16384
ip-route             80000      262144     80000
ip-static-route      64         2048       64
vlan                 64         4095       64
spanning-tree        32         255        32
mac-filter-port      16         256        16
mac-filter-sys       32         512        32
ip-subnet-port       24         128        24
session-limit        65536      160000     65536
view                 10         65535      10
virtual-interface    255        512        255
hw-ip-next-hop       6144       6144       6144
hw-ip-mcast-mll      3072       3072       3072
hw-traffic-condition 50         1024       50
rmon-entries         2048      32768       2048
```

Table 8.1 defines the system parameters in the **show default values** command output.

Table 8.1: System Parameters in Show Default Values Command

This system parameter...	Defines the maximum number of...
atalk-route	Appletalk routes
atalk-zone-port	Appletalk zones per port
atalk-zone-sys	Appletalk zones per system
dvmrp-mcache	PIM and DVMRP multicast cache flows stored in CAM
dvmrp-route	DVMRP routes
hw-ip-mcast-mll	Multicast output interfaces (clients)
hw-ip-next-hop	IP next hops and routes, including unicast next hops and multicast route entries
hw-logical-interface	Hardware logical interface pairs (physical port and VLAN pairs)
hw-traffic-conditioner	Traffic policies
ip-arp	ARP entries
ip-cache	IP forwarding cache entries
ip-filter-port	IP ACL entries per port
ip-filter-sys	IP ACL entries per system
ip-qos-session	Layer 4 session table entries
ip-route	Learned IP routes
ip-static-arp	Static IP ARP entries
ip-static-route	Static IP routes
ip-subnet-port	IP subnets per port
ipx-forward-filter	IPX forward filter entries
ipx-rip-entry	IPX RIP entries
ipx-rip-filter	IPX RIP filter entries
ipx-sap-entry	IPX SAP entries
ipx-sap-filter	IPX SAP filter entries
l3-vlan	Layer 3 VLANs
mac	MAC entries
mac-filter-port	Layer 2 MAC filter entries per port
mac-filter-sys	Layer 2 MAC filter entries per system
multicast-route	Multicast routes
pim-mcache	PIM multicast cache entries

Table 8.1: System Parameters in Show Default Values Command (Continued)

This system parameter...	Defines the maximum number of...
rmon-entries	RMON control table entries
session-limit	Session entries
spanning-tree	Spanning tree instances
view	SNMP views
virtual-interface	Virtual routing interfaces
vlan	VLANs

Modifying System Parameter Default Values

Information for the configurable tables appears under the columns that are shown in bold type in the above examples. To simplify configuration, the command parameter you enter to configure the table is used for the table name. For example, to increase the capacity of the IP route table, enter the following commands:

```
FastIron(config)#system-max ip-route 120000
FastIron(config)#write memory
FastIron(config)#exit
FastIron#reload
```

Syntax: system-max ip-route <num>

The <num> parameter specifies the maximum number of routes in the IP route table. This value can be from 4096 – 128000. For FastIron chassis devices running software release 02.5.00 or later, this value can be from 4096 – 262144. The default is 80000 IP routes.

NOTE: If you accidentally enter a value that is not within the valid range of values, the CLI will display the valid range for you.

To increase the number of IP subnet interfaces you can configure on each port on a device running Layer 3 code from 24 to 64, then increase the total number of IP interfaces you can configure on the device from 256 to 512, enter the following commands:

```
FastIron(config)#system-max subnet-per-interface 64
FastIron(config)#write memory
FastIron(config)#exit
FastIron#reload
```

Syntax: system-max subnet-per-interface <num>

The <num> parameter specifies the maximum number of subnet addresses per port and can be from 1 – 64. The default is 24.

```
FastIron(config)#system-max subnet-per-system 512
FastIron(config)#write memory
FastIron(config)#exit
FastIron#reload
```

Syntax: system-max subnet-per-system <num>

The <num> parameter specifies the maximum number of subnet addresses for the entire device and can be from 1 – 512. The default is 256.

NOTE: If you increase the number of configurable subnet addresses on each port, you might also need to increase the total number of subnets that you can configure on the device.

Dynamic Buffer Allocation

Platform Support:

- FESX/FSX/FWSX devices running software release 02.4.00 and later
- FGS and FLS devices running software releases 03.0.00 and later

The Foundry device allocates a certain number of buffers to each port's outbound transmit queue based on priority (traffic class). The buffers control the total number of packets permitted in the port's outbound transmit queue. For each port, the Foundry device defines the maximum outbound transmit buffers, also called *queue depth limits*, as follows:

- ***Total Transmit Queue Depth Limit*** – The total maximum number of transmit buffers allocated for all outbound packets on a port. Packets are added to the port's outbound queue as long as the number of buffers currently being used is less than the total transmit queue depth limit. When this limit is reached, any new packets attempting to enter the port's transmit queue will be dropped until at least one buffer is freed.
- ***Transmit Queue Depth Limit for a Given Traffic Class*** – The maximum number of transmit buffers allocated for packets with a given traffic class (0 – 7) on a port. Packets with the specified traffic class are added to the port's outbound queue as long as the number of buffers currently being used for that traffic class is less than the transmit queue depth limit for the traffic class. When this limit is reached, any new packets with the specified traffic class attempting to enter the port's transmit queue will be dropped.

In releases prior to 02.4.00 for the FastIron X Series, the above queue depth limits are fixed and cannot be modified.

Starting with release 02.4.00 for the FastIron X Series, you can increase or decrease both of these queue depth limits per port. This feature is useful in situations where applications have intermittent bursts of oversubscription. For example, by increasing the buffers on the egress port, the Foundry device will be able to forward oversubscribed packets instead of dropping them.

Foundry devices use the default maximum queue depths listed in Table 8.2.

Default Queue Depth Limits

Table 8.2 defines the default maximum queue depth values per port, per traffic class. The Foundry device drops the packets that cause the port to exceed these limits.

Table 8.2: Default Maximum Queue Depth

Port Type	Maximum Queue Depth Per Port, Per Priority 0 – 6	Maximum Queue Depth Per Port, Per Priority 7	Total Maximum Queue Depth
1-Gigabit port	96	224	896
10-Gigabit port without jumbo enabled	400	1104	3904
10-Gigabit port with jumbo enabled	352	640	3104

Configuring the Total Transmit Queue Depth Limit

To set the total transmit queue depth limit on a port, enter a command such as the following:

```
FastIron(config)#qd 2 2049
```

This command sets the queue depth limit on port 2 to 2049. Packets are added to the port's outbound queue as long as the packets do not cause the port to exceed 2048 buffers. If the port reaches its queue depth limit of 2049, any new packets attempting to enter the port’s transmit queue will be dropped until at least one buffer is freed.

Syntax: qd [<slotnum>/]<portnum> <limit>

The <slotnum> parameter is required on chassis devices.

The <portnum> parameter specifies on which port to apply the queue depth limit.

The <limit> parameter can be a value from 0 – 4095. Table 8.2 lists the default values.

Configuring the Transmit Queue Depth Limit for a Given Traffic Class

To set the transmit queue depth limit on a port for a given traffic class, first enter the transmit queue depth limit for the traffic class, then specify the traffic class. For example:

```
FastIron(config)#qd 2 200 7
```

This command sets the queue depth limit on port 2 to 200 for packets with a traffic class of 7. Packets with priority 7 are added to port 2’s outbound queue as long as the packets do not exceed 199 buffers. When the port reaches its queue depth limit of 200, packets with the given traffic class will be dropped.

Syntax: qd [<slotnum>/]<portnum> <limit> <traffic-class>

The <slotnum> parameter is required on chassis devices.

The <portnum> parameter specifies on which port to apply the queue depth limit.

The <limit> parameter can be a value from 0 – 4095 and cannot exceed the port’s configured total transmit queue depth limit. Table 8.2 lists the default values.

- The sum of the queue depth limits for individual traffic classes on a port does not need to equal the total queue depth limit for the port.
    - If the sum of the individual traffic class queue depth limits exceeds the total port limit and the total port limit is reached, any buffer that gets released can be used by any traffic class queue that has not reached its individual limit.
    - If the sum of the individual traffic class queue depth limits is less than the total port limit, the remaining buffers can be used only by packets with a priority of 7.

The <traffic-class> parameter can be a value from 0 – 7, where 7 is the highest priority queue.

Configuring the Transmit Queue Depth Limit for a Given Traffic Class

To set the transmit queue depth limit on a port for a given traffic class, first enter the transmit queue depth limit for the traffic class, then specify the traffic class. For example:

```
FastIron(config)#qd 2 200 7
```

This command sets the queue depth limit on port 2 to 200 for packets with a traffic class of 7. Packets with priority 7 are added to port 2’s outbound queue as long as the packets do not exceed 199 buffers. When the port reaches its queue depth limit of 200, packets with the given traffic class will be dropped.

Syntax: qd [<slotnum>/]<portnum> <limit> <traffic-class>

The <slotnum> parameter is required on chassis devices.

The <portnum> parameter specifies on which port to apply the queue depth limit.

The <limit> parameter can be a value from 0 – 4095 and cannot exceed the port’s configured total transmit queue depth limit. Table 8.2 lists the default values.

- The sum of the queue depth limits for individual traffic classes on a port does not need to equal the total queue depth limit for the port.
    - If the sum of the individual traffic class queue depth limits exceeds the total port limit and the total port limit is reached, any buffer that gets released can be used by any traffic class queue that has not reached its individual limit.
    - If the sum of the individual traffic class queue depth limits is less than the total port limit, the remaining buffers can be used only by packets with a priority of 7.

The <traffic-class> parameter can be a value from 0 – 7, where 7 is the highest priority queue.

Removing Buffer Allocation Limits

Platform Support:

- FESX/FSX/FWSX devices running software release 03.2.00 and later

You can remove buffer allocation limits on all ports and all Traffic Classes globally. This permits all available buffers in a port region to be used in a first-come-first-serve basis by any of its ports, regardless of priority. This can be done using the following command:

```
FastIron(config)#buffer-sharing-full
```

Syntax: [no] buffer-sharing-full

Entering this command sets the Total Transmit Queue Depth Limit as well as the Transmit Queue Depth Limits for each Traffic Class to 4095 for all ports of the device. The command overrides any existing individually configured queue depth limits.

NOTE: This command should be used carefully! By entering this command, there is no limit on the number of buffers a port or a specific priority on a port can use. So one port could potentially use up all the available buffers of its port region and cause starvation on other ports of the port region.

Generic Buffer Profiles

Platform Support:

- FGS and FLS devices running software release 04.1.00 and later

Default buffer settings are currently optimized for 1GE-to-1GE traffic. This feature adds buffer profiles for 1GE-to-100Mbit traffic, simplifying configuration and improving performance.

This feature allows users to configure a pre-defined set of buffers and descriptors for priority 0 and 7. The buffer profile supports VoIP traffic that uses priority 7, with 10/100 uplink ports and 1000 downlink ports.

NOTE: In previous versions, users could manually configure buffers and descriptors using QD commands. This feature cannot co-exist with QD commands. You may use one or the other, but not both types at the same time.

Configuring Buffer Profiles

To configure predefined buffers, enter a command similar to the following:

```
FastIron#buffer-profile port-region 0 voip downlink 100 uplink 1000
```

Syntax: [no] buffer-profile port-region <num> voip downlink 100 uplink 1000

NOTE: The port-region num can be either 0 (ports 0/1/1 to 0/1/24) or 1 (ports 0/1/25 to 0/1/48).

Deleting Buffer Profiles

To delete an existing buffer profile configuration, use the [no] form of the command:

```
FastIron#no buffer-profile port-region 0 voip downlink 100 uplink 1000
```

Remote Fault Notification (RFN) on 1G Fiber Connections

Platform Support:

- FGS and FLS devices running software release 02.6.00 and later
- FESX/FSX/FWSX devices – all sofware releases

NOTE: This feature is only available for Gigabit Ethernet Fiber ports. It is not available for 10/100 ports and Gigabit Ethernet Copper ports.

For fiber-optic connections, you can optionally configure a transmit port to notify the receive port on the remote device whenever the transmit port becomes disabled.

When you enable this feature, the transmit port notifies the remote port whenever the fiber cable is either physically disconnected or has failed. When this occurs and the feature is enabled, the device disables the link and turns OFF both LEDs associated with the ports.

By default, RFN is enabled.

You can configure RFN as follows:

- Globally, on the entire device
- On a trunk group
- On an individual interface

Enabling and Disabling Remote Fault Notification

RFN is ON by default. To disable RFN, use the following command:

```
FastIron(config)#interface e 0/1/1
FastIron(config-if-e1000-0/1/1)#gig-default neg-off
```

To re-enable RFN, use the following command:

```
FastIron(config)#interface e 0/1/1
FastIron(config-if-e1000-0/1/1)#gig-default auto-on
```

Link Fault Signaling (LFS) for 10G

Platform Support:

- FGS and FLS devices running software release 02.6.00 and later
- FESX/FSX/FWSX devices – all sofware releases

Link Fault Signaling (LFS) is a physical layer protocol that enables communication on a link between two 10 Gigabit Ethernet devices. When configured on a Foundry 10 Gigabit Ethernet port, the port can detect and report fault conditions on transmit and receive ports. Foundry recommends enabling LFS on both ends of a link.

When LFS is enabled on an interface, the following Syslog messages are generated when the link goes up or down or when the TX or RX fiber is removed from one or both sides of the link that has LFS enabled:

```
Interface ethernet1/1, state down - link down
```

```
Interface ethernet1/1, state up
```

When a link fault occurs, the Link and Activity LEDs turn OFF.

The Link and Activity LEDs turn ON when there is traffic traversing the link after the fiber is installed.

Enabling LFS

To enable LFS between two 10 Gigabit Ethernet devices, enter commands such as the following on both ends of the link:

```
FastIron(config)#interface e 1/1
FastIron(config-if-e1000-1/1)#link-fault-signal
```

Syntax: [no] link-fault-signal

Use the no form of the command to disable LFS.

LFS is OFF by default.

Chapter 9

Configuring Spanning Tree Protocol (STP) Related Features

This chapter describes how to configure Spanning Tree Protocol (STP) and other STP parameters on Foundry Layer 3 Switches using the CLI. STP related features, such as RSTP and PVST, extend the operation of standard STP, enabling you to fine tune standard STP and avoid some of its limitations.

STP Overview

The Spanning Tree Protocol (STP) eliminates Layer 2 loops in networks, by selectively blocking some ports and allowing other ports to forward traffic, based on global (bridge) and local (port) parameters you can configure.

You can enable or disable STP on a global basis (for the entire device), a port-based VLAN basis (for the individual Layer 2 broadcast domain), or an individual port basis.

Configuration procedures are provided for the standard STP bridge and port parameters as well as Foundry features listed in Table 9.6.

Configuring Standard STP Parameters

Foundry Layer 2 Switches and Layer 3 Switches support standard STP as described in the IEEE 802.1D specification. STP is enabled by default on Layer 2 Switches but disabled by default on Layer 3 Switches.

By default, each port-based VLAN on a Foundry device runs a separate spanning tree (a separate instance of STP). A Foundry device has one port-based VLAN (VLAN 1) by default that contains all the device’s ports. Thus, by default each Foundry device has one spanning tree. However, if you configure additional port-based VLANs on a Foundry device, then each of those VLANs on which STP is enabled and VLAN 1 all run separate spanning trees.

If you configure a port-based VLAN on the device, the VLAN has the same STP state as the default STP state on the device. Thus, on Layer 2 Switches, new VLANs have STP enabled by default. On Layer 3 Switches, new VLANs have STP disabled by default. You can enable or disable STP in each VLAN separately. In addition, you can enable or disable STP on individual ports.

STP Parameters and Defaults

Table 9.1 lists the default STP states for Foundry devices.

Table 9.1: Default STP States

Device Type	Default STP Type	Default STP State	Default STP State of New VLANs[1]
Layer 2 Switch	MSTP[2]	Enabled	Enabled
Layer 3 Switch	MSTP	Disabled	Disabled

1. When you create a port-based VLAN, the new VLAN's STP state is the same as the default STP state on the device. The new VLAN does not inherit the STP state of the default VLAN.
2. MSTP stands for "Multiple Spanning Tree Protocol". In this type of STP, each port-based VLAN, including the default VLAN, has its own spanning tree. References in this documentation to "STP" apply to MSTP. The Single Spanning Tree Protocol (SSTP) is another type of STP. SSTP includes all VLANs on which STP is enabled in a single spanning tree. See "Single Spanning Tree (SSTP)" on page 9-57.

Table 9.2 lists the default STP bridge parameters. The bridge parameters affect the entire spanning tree. If you are using MSTP, the parameters affect the VLAN. If you are using SSTP, the parameters affect all VLANs that are members of the single spanning tree.

Table 9.2: Default STP Bridge Parameters

Parameter	Description	Default and Valid Values
Forward Delay	The period of time spent by a port in the listening and learning state before moving on to the learning or forwarding state, respectively. The forward delay value is also used for the age time of dynamic entries in the filtering database, when a topology change occurs.	15 seconds Possible values: 4 – 30 seconds
Maximum Age	The interval a bridge will wait for a configuration BPDU from the root bridge before initiating a topology change.	20 seconds Possible values: 6 – 40 seconds
Hello Time	The interval of time between each configuration BPDU sent by the root bridge.	2 seconds Possible values: 1 – 10 seconds
Priority	A parameter used to identify the root bridge in a spanning tree (instance of STP). The bridge with the lowest value has the highest priority and is the root. A higher numerical value means a lower priority; thus, the highest priority is 0.	32768 Possible values: 0 – 65535

NOTE: If you plan to change STP bridge timers, Foundry recommends that you stay within the following ranges, from section 8.10.2 of the IEEE STP specification.

2 * (forward_delay -1) >= max_age

max_age >= 2 * (hello_time +1)

Table 9.3 lists the default STP port parameters. The port parameters affect individual ports and are separately configurable on each port.

Table 9.3: Default STP Port Parameters

Parameter	Description	Default and Valid Values
Priority	The preference that STP gives this port relative to other ports for forwarding traffic out of the spanning tree. A higher numerical value means a lower priority.	128 Possible values in FastIron X Series pre-release 03.0.00, and in FGS pre-release 03.0.00: 8 – 252 (configurable in increments of 4) Possible values in FastIron X Series and FGS/FLS release 03.0.00 and later: 0 – 240 (configurable in increments of 16)
Path Cost	The cost of using the port to reach the root bridge. When selecting among multiple links to the root bridge, STP chooses the link with the lowest path cost and blocks the other paths. Each port type has its own default STP path cost.	10 Mbps – 100 100 Mbps – 19 Gigabit – 4 10 Gigabit – 2 Possible values are 0 – 65535

Enabling or Disabling the Spanning Tree Protocol (STP)

STP is *enabled* by default on devices running Layer 2 code. STP is *disabled* by default on devices running Layer 3 code.

You can enable or disable STP on the following levels:

- Globally – Affects all ports and port-based VLANs on the device.
- Port-based VLAN – Affects all ports within the specified port-based VLAN. When you enable or disable STP within a port-based VLAN, the setting overrides the global setting. Thus, you can enable STP for the ports within a port-based VLAN even when STP is globally disabled, or disable the ports within a port-based VLAN when STP is globally enabled.
- Individual port – Affects only the individual port. However, if you change the STP state of the primary port in a trunk group, the change affects all ports in the trunk group.

NOTE: The CLI converts the STP groups into topology groups when you save the configuration. For backward compatibility, you can still use the STP group commands. However, the CLI converts the commands into the topology group syntax. Likewise, the **show stp-group** command displays STP topology groups. "Topology Groups" on page 10-1

Enabling or Disabling STP Globally

Use the following method to enable or disable STP on a device on which you have not configured port-based VLANs.

NOTE: When you configure a VLAN, the VLAN inherits the global STP settings. However, once you begin to define a VLAN, you can no longer configure standard STP parameters globally using the CLI. From that point on, you can configure STP only within individual VLANs.

To enable STP for all ports in all VLANs on a Foundry device, enter the following command:

```
FastIron(config)#spanning-tree
```

This command enables a separate spanning tree in each VLAN, including the default VLAN.

Syntax: [no] spanning-tree

Enabling or Disabling STP in a Port-Based VLAN

Use the following procedure to disable or enable STP on a device on which you have configured a port-based VLAN. Changing the STP state in a VLAN affects only that VLAN.

To enable STP for all ports in a port-based VLAN, enter commands such as the following:

```
FastIron(config)#vlan 10
FastIron(config-vlan-10)#spanning-tree
```

Syntax: [no] spanning-tree

Enabling or Disabling STP on an Individual Port

Use the following procedure to disable or enable STP on an individual port.

NOTE: If you change the STP state of the primary port in a trunk group, it affects all ports in the trunk group.

To enable STP on an individual port, enter commands such as the following:

```
FastIron(config)#interface 1/1
FastIron(config-if-e1000-1/1)#spanning-tree
```

Syntax: [no] spanning-tree

Changing STP Bridge and Port Parameters

Table 9.2 on page 9-2 and Table 9.3 on page 9-3 list the default STP parameters. If you need to change the default value for an STP parameter, use the following procedures.

Changing STP Bridge Parameters

NOTE: If you plan to change STP bridge timers, Foundry recommends that you stay within the following ranges, from section 8.10.2 of the IEEE STP specification.

2 * (forward_delay -1) >= max_age

max_age >= 2 * (hello_time +1)

To change a Foundry device's STP bridge priority to the highest value to make the device the root bridge, enter the following command:

```
FastIron(config)#spanning-tree priority 0
```

The command in this example changes the priority on a device on which you have not configured port-based VLANs. The change applies to the default VLAN. If you have configured a port-based VLAN on the device, you can configure the parameters only at the configuration level for individual VLANs. Enter commands such as the following:

```
FastIron(config)#vlan 20
FastIron(config-vlan-20)#spanning-tree priority 0
```

To make this change in the default VLAN, enter the following commands:

```
FastIron(config)#vlan 1
FastIron(config-vlan-1)#spanning-tree priority 0
```

Syntax: [no] spanning-tree [forward-delay <value>] | [hello-time <value>] | [maximum-age <value>] | [priority <value>]

The **forward-delay** <value> parameter specifies the forward delay and can be a value from 4 – 30 seconds. The default is 15 seconds.

NOTE: You can configure a Foundry device for faster convergence (including a shorter forward delay) using Fast Span. See "Configuring STP Related Features" on page 9-16.

The **hello-time** <value> parameter specifies the hello time and can be a value from 1 – 10 seconds. The default is 2 seconds.

NOTE: This parameter applies only when this device or VLAN is the root bridge for its spanning tree.

The **maximum-age** <value> parameter specifies the amount of time the device waits for receipt of a configuration BPDU from the root bridge before initiating a topology change. You can specify from 6 – 40 seconds. The default is 20 seconds.

The **priority** <value> parameter specifies the priority and can be a value from 0 – 65535. A higher numerical value means a lower priority. Thus, the highest priority is 0. The default is 32768.

You can specify some or all of these parameters on the same command line. If you specify more than one parameter, you must specify them in the order shown above, from left to right.

Changing STP Port Parameters

To change the path and priority costs for a port, enter commands such as the following:

```
FastIron(config)#vlan 10
FastIron(config-vlan-10)#spanning-tree ethernet 5 path-cost 15 priority 64
```

Syntax: spanning-tree ethernet [<slotnum>/]<portnum> path-cost <value> | priority <value> | disable | enable

The <portnum> parameter specifies the interface. If you are configuring a chassis device, specify the slot number as well as the port number (<slotnum>/<portnum>).

The **path-cost** <value> parameter specifies the port's cost as a path to the spanning tree's root bridge. STP prefers the path with the lowest cost. You can specify a value from 0 – 65535.

The default depends on the port type:

- 10 Mbps – 100
- 100 Mbps – 19
- Gigabit – 4

- 10 Gigabit – 2

The **priority** <value> parameter specifies the preference that STP gives this port relative to other ports for forwarding traffic out of the spanning tree. The value you can specify depends on the software version running on the device, as follows:

- In releases prior to 03.0.00, you can specify a value from 8 – 252, in increments of 4. If you enter a value that is not divisible by four the software rounds to the nearest value that is. The default is 128. A higher numerical value means a lower priority; thus, the highest priority is 8.
- Starting in software release 03.0.00, you can specify a value from 0 – 240, in increments of 16. If you enter a value that is not divisible by 16, the software returns an error message. The default value is 128. A higher numerical value means a lower priority; thus, the highest priority is 0.

NOTE: If you are upgrading a device that has a configuration saved under an earlier software release, and the configuration contains a value from 0 – 7 for a port's STP priority, the software changes the priority to the default when you save the configuration while running the new release.

The **disable | enable** parameter disables or re-enables STP on the port. The STP state change affects only this VLAN. The port's STP state in other VLANs is not changed.

STP Protection Enhancement

Platform Support:

- FESX devices running software release 02.1.01 and later
- FSX/FWSX devices running software release 02.0.00 and later

STP protection provides the ability to prohibit an end station from initiating or participating in an STP topology change.

The 802.1W Spanning Tree Protocol (STP) detects and eliminates logical loops in a redundant network by selectively blocking some data paths (ports) and allowing only the best data paths to forward traffic.

In an STP environment, switches, end stations, and other Layer 2 devices use Bridge Protocol Data Units (BPDUs) to exchange information that STP will use to determine the best path for data flow. When a Layer 2 device is powered ON and connected to the network, or when a Layer 2 device goes down, it sends out an STP BPDU, triggering an STP topology change.

In some instances, it is unnecessary for a connected device, such as an end station, to initiate or participate in an STP topology change. In this case, you can enable the STP Protection feature on the Foundry port to which the end station is connected. Foundry's STP Protection feature disables the connected device's ability to initiate or participate in an STP topology change, by dropping all BPDUs received from the connected device.

Enabling STP Protection

You can enable STP Protection on a per-port basis.

To prevent an end station from initiating or participating in STP topology changes, enter the following command at the Interface level of the CLI:

```
FastIron#(config) interface e 2
FastIron#(config-if-e1000-2)#stp-protect
```

This command causes the port to drop STP BPDUs sent from the device on the other end of the link.

Syntax: [no] stp-protect

Enter the **no** form of the command to disable STP protection on the port.

Clearing BPDU Drop Counters

For each port that has STP Protection enabled, the Foundry device counts and records the number of dropped BPDUs. You can use CLI commands to clear the BPDU drop counters for all ports on the device, or for a specific port on the device.

To clear the BPDU drop counters for all ports on the device that have STP Protection enabled, enter the following command at the Global CONFIG level of the CLI:

```
FastIron(config)#clear stp-protect-statistics
```

To clear the BPDU drop counter for a specific port that has STP Protection enabled, enter the following command at the Global CONFIG level of the CLI:

```
FastIron#clear stp-protect-statistics e 2
```

Syntax: clear stp-protect-statistics [ethernet [<slotnum>/]<port-num>] | [ethernet [<slotnum>/]portnum>]

Viewing the STP Protection Configuration

You can view the STP Protection configuration for all ports on a device, or for a specific port only. The **show stp-protect** command output shows the port number on which STP Protection is enabled, and the number of BPDUs dropped by each port.

To view the STP Protection configuration for all ports on the device, enter the following command at any level of the CLI:

```
FastIron#show stp-protect
Port ID        BPDU Drop Count
  3                  478
  5                  213
  6                    0
  12                  31
```

To view STP Protection configuration for a specific port, enter the following command at any level of the CLI:

```
FastIron#show stp-protect e 3
STP-protect is enabled on port 3.  BPDU drop count is 478
```

If you enter the **show stp-protect** command for a port that does not have STP protection enabled, the following message displays on the console:

```
FastIron#show stp-protect e 4
STP-protect is not enabled on port 4.
```

Syntax: show stp-protect [ethernet [<slotnum>/]<portnum>]

Displaying STP Information

You can display the following STP information:

- All the global and interface STP settings
- CPU utilization statistics
- Detailed STP information for each interface
- STP state information for a port-based VLAN
- STP state information for an individual interface

Displaying STP Information for an Entire Device

To display STP information, enter the following command at any level of the CLI:

```
FastIron#show span

VLAN 1 BPDU cam_index is 3 and the Master DMA Are(HEX)
STP instance owned by VLAN 1

Global STP (IEEE 802.1D) Parameters:

VLAN Root             Root Root Prio Max He- Ho- Fwd Last      Chg  Bridge
 ID   ID              Cost Port rity Age llo ld  dly Chang     cnt  Address
                                Hex  sec sec sec sec sec
   1 800000e0804d4a00 0    Root 8000 20  2   1   15  689       1    00e0804d4a00

Port STP Parameters:

Port  Prio Path  State      Fwd    Design   Designated       Designated
Num   rity Cost             Trans  Cost     Root             Bridge
      Hex
1     80   19    FORWARDING 1      0        800000e0804d4a00 800000e0804d4a00
2     80   0     DISABLED   0      0        0000000000000000 0000000000000000
3     80   0     DISABLED   0      0        0000000000000000 0000000000000000
4     80   0     DISABLED   0      0        0000000000000000 0000000000000000
5     80   19    FORWARDING 1      0        800000e0804d4a00 800000e0804d4a00
6     80   19    BLOCKING   0      0        800000e0804d4a00 800000e0804d4a00
7     80   0     DISABLED   0      0        0000000000000000 0000000000000000
```

<lines for remaining ports excluded for brevity>

Syntax: show span [vlan <vlan-id>] | [pvst-mode] | [<num>] | [detail [vlan <vlan-id> [ethernet [<slotnum>/]<portnum>] | <num>]]

The **vlan** <vlan-id> parameter displays STP information for the specified port-based VLAN.

The **pvst-mode** parameter displays STP information for the device's Per VLAN Spanning Tree (PVST+) compatibility configuration. See "PVST/PVST+ Compatibility" on page 9-62.

The <num> parameter displays only the entries after the number you specify. For example, on a device with three port-based VLANs, if you enter 1, then information for the second and third VLANs is displayed, but information for the first VLAN is not displayed. Information is displayed according to VLAN number, in ascending order. The entry number is not the same as the VLAN number. For example, if you have port-based VLANs 1, 10, and 2024, then the command output has three STP entries. To display information for VLANs 10 and 2024 only, enter **show span 1**.

The **detail** parameter and its additional optional parameters display detailed information for individual ports. See "Displaying Detailed STP Information for Each Interface" on page 9-12.

The **show span** command shows the following information.

Table 9.4: CLI Display of STP Information

This Field...	Displays...
Global STP Parameters	
VLAN ID	The port-based VLAN that contains this spanning tree (instance of STP). VLAN 1 is the default VLAN. If you have not configured port-based VLANs on this device, all STP information is for VLAN 1.
Root ID	The ID assigned by STP to the root bridge for this spanning tree.
Root Cost	The cumulative cost from this bridge to the root bridge. If this device is the root bridge, then the root cost is 0.
Root Port	The port on this device that connects to the root bridge. If this device is the root bridge, then the value is “Root” instead of a port number.
Priority Hex	This device or VLAN’s STP priority. The value is shown in hexadecimal format. **Note**: If you configure this value, specify it in decimal format. See “Changing STP Bridge Parameters” on page 9-4.
Max age sec	The number of seconds this device or VLAN waits for a configuration BPDU from the root bridge before deciding the root has become unavailable and performing a reconvergence.
Hello sec	The interval between each configuration BPDU sent by the root bridge.
Hold sec	The minimum number of seconds that must elapse between transmissions of consecutive Configuration BPDUs on a port.
Fwd dly sec	The number of seconds this device or VLAN waits following a topology change and consequent reconvergence.
Last Chang sec	The number of seconds since the last time a topology change occurred.
Chg cnt	The number of times the topology has changed since this device was reloaded.
Bridge Address	The STP address of this device or VLAN. **Note**: If this address is the same as the Root ID, then this device or VLAN is the root bridge for its spanning tree.
Port STP Parameters	
Port Num	The port number.
Priority Hex	The port’s STP priority, in hexadecimal format. **Note**: If you configure this value, specify it in decimal format. See “Changing STP Port Parameters” on page 9-5.
Path Cost	The port’s STP path cost.

Table 9.4: CLI Display of STP Information (Continued)

This Field...	Displays...
State	The port's STP state. The state can be one of the following: • BLOCKING – STP has blocked Layer 2 traffic on this port to prevent a loop. The device or VLAN can reach the root bridge using another port, whose state is FORWARDING. When a port is in this state, the port does not transmit or receive user frames, but the port does continue to receive STP BPDUs. • DISABLED – The port is not participating in STP. This can occur when the port is disconnected or STP is disabled on the port. • FORWARDING – STP is allowing the port to send and receive frames. • LISTENING – STP is responding to a topology change and this port is listening for a BPDU from neighboring bridge(s) in order to determine the new topology. No user frames are transmitted or received during this state. • LEARNING – The port has passed through the LISTENING state and will change to the FORWARDING state, depending on the results of STP's reconvergence. The port does not transmit or receive user frames during this state. However, the device can learn the MAC addresses of frames that the port receives during this state and make corresponding entries in the MAC table.
Fwd Trans	The number of times STP has changed the state of this port between BLOCKING and FORWARDING.
Design Cost	The cost to the root bridge as advertised by the designated bridge that is connected to this port. If the designated bridge is the root bridge itself, then the cost is 0. The identity of the designated bridge is shown in the Design Bridge field.
Designated Root	The root bridge as recognized on this port. The value is the same as the root bridge ID listed in the Root ID field.
Designated Bridge	The designated bridge to which this port is connected. The designated bridge is the device that connects the network segment on the port to the root bridge.

Displaying CPU Utilization Statistics

You can display CPU utilization statistics for STP and the IP protocols.

To display CPU utilization statistics for STP for the previous one-second, one-minute, five-minute, and fifteen-minute intervals, enter the following command at any level of the CLI:

```
FastIron#show process cpu
Process Name   5Sec(%)   1Min(%)   5Min(%)   15Min(%)   Runtime(ms)
ARP              0.01      0.03      0.09      0.22              9
BGP              0.04      0.06      0.08      0.14              13
GVRP             0.00      0.00      0.00      0.00              0
ICMP             0.00      0.00      0.00      0.00              0
IP               0.00      0.00      0.00      0.00              0
OSPF             0.00      0.00      0.00      0.00              0
RIP              0.00      0.00      0.00      0.00              0
STP              0.00      0.03      0.04      0.07              4
VRRP             0.00      0.00      0.00      0.00              0
```

If the software has been running less than 15 minutes (the maximum interval for utilization statistics), the command indicates how long the software has been running. Here is an example:

```
FastIron#show process cpu
The system has only been up for 6 seconds.
Process Name   5Sec(%)   1Min(%)   5Min(%)   15Min(%)   Runtime(ms)
ARP              0.01      0.00      0.00      0.00               0
BGP              0.00      0.00      0.00      0.00               0
GVRP             0.00      0.00      0.00      0.00               0
ICMP             0.01      0.00      0.00      0.00               1
IP               0.00      0.00      0.00      0.00               0
OSPF             0.00      0.00      0.00      0.00               0
RIP              0.00      0.00      0.00      0.00               0
STP              0.00      0.00      0.00      0.00               0
VRRP             0.00      0.00      0.00      0.00               0
```

To display utilization statistics for a specific number of seconds, enter a command such as the following:

```
FastIron#show process cpu 2
Statistics for last 1 sec and 80 ms
Process Name   Sec(%)   Time(ms)
ARP              0.00        0
BGP              0.00        0
GVRP             0.00        0
ICMP             0.01        1
IP               0.00        0
OSPF             0.00        0
RIP              0.00        0
STP              0.01        0
VRRP             0.00        0
```

When you specify how many seconds' worth of statistics you want to display, the software selects the sample that most closely matches the number of seconds you specified. In this example, statistics are requested for the previous two seconds. The closest sample available is actually for the previous 1 second plus 80 milliseconds.

Syntax: show process cpu [<num>]

The <num> parameter specifies the number of seconds and can be from 1 – 900. If you use this parameter, the command lists the usage statistics only for the specified number of seconds. If you do not use this parameter, the command lists the usage statistics for the previous one-second, one-minute, five-minute, and fifteen-minute intervals.

Displaying the STP State of a Port-Based VLAN

When you display information for a port-based VLAN, that information includes the STP state of the VLAN.

To display information for a port-based VLAN, enter a command such as the following at any level of the CLI. The STP state is shown in bold type in this example.

```
FastIron#show vlans

Total PORT-VLAN entries: 2
Maximum PORT-VLAN entries: 16

legend: [S=Slot]

PORT-VLAN 1, Name DEFAULT-VLAN, Priority level0, Spanning tree On
 Untagged Ports: (S3)  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16
 Untagged Ports: (S3) 17 18 19 20 21 22 23 24
 Untagged Ports: (S4)  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17
 Untagged Ports: (S4) 18 19 20 21 22 23 24
   Tagged Ports: None
   Uplink Ports: None

PORT-VLAN 2, Name greenwell, Priority level0, Spanning tree Off
 Untagged Ports: (S1)  1  2  3  4  5  6  7  8
 Untagged Ports: (S4)  1
   Tagged Ports: None
   Uplink Ports: None
```

Syntax: show vlans [<vlan-id> | ethernet [<slotnum>/]<portnum>

The <vlan-id> parameter specifies a VLAN for which you want to display the configuration information.

The <portnum> parameter specifies a port. If you use this parameter, the command lists all the VLAN memberships for the port. If you use this command on a chassis device, specify the slot number as well as the port number (<slotnum>/]<portnum>).

Displaying Detailed STP Information for Each Interface

To display the detailed STP information, enter the following command at any level of the CLI:

```
FastIron#show span detail
======================================================================
VLAN 1 - MULTIPLE SPANNING TREE (MSTP) ACTIVE
======================================================================
Bridge identifier    - 0x800000e0804d4a00
Active global timers - Hello: 0

Port 1/1 is FORWARDING
  Port - Path cost: 19, Priority: 128, Root: 0x800000e052a9bb00
  Designated - Bridge: 0x800000e052a9bb00, Interface: 1, Path cost: 0
  Active Timers - None
  BPDUs - Sent: 11, Received: 0
Port 1/2 is DISABLED
Port 1/3 is DISABLED
Port 1/4 is DISABLED
<lines for remaining ports excluded for brevity>
```

If a port is disabled, the only information shown by this command is “DISABLED”. If a port is enabled, this display shows the following information.

Syntax: show span detail [vlan <vlan-id> [ethernet [<slotnum>/]<portnum> | <num>]

The **vlan** <vlan-id> parameter specifies a VLAN.

The <portnum> parameter specifies an individual port within the VLAN (if specified). If you use the command on a chassis device, specify the slot number as well as the port number (<slotnum>/<portnum>).

The <num> parameter specifies the number of VLANs you want the CLI to skip before displaying detailed STP information. For example, if the device has six VLANs configured (VLAN IDs 1, 2, 3, 99, 128, and 256) and you enter the command **show span detail 4**, detailed STP information is displayed for VLANs 128 and 256 only.

NOTE: If the configuration includes VLAN groups, the **show span detail** command displays the master VLANs of each group but not the member VLANs within the groups. However, the command does indicate that the VLAN is a master VLAN. The **show span detail vlan** <vlan-id> command displays the information for the VLAN even if it is a member VLAN. To list all the member VLANs within a VLAN group, enter the **show vlan-group** [<group-id>] command.

The **show span detail** command shows the following information.

Table 9.5: CLI Display of Detailed STP Information for Ports

This Field...	Displays...
Active Spanning Tree protocol	The VLAN that contains the listed ports and the active Spanning Tree protocol. The STP type can be one of the following: • MULTIPLE SPANNNG TREE (MSTP) • GLOBAL SINGLE SPANNING TREE (SSTP) **Note**: If STP is disabled on a VLAN, the command displays the following message instead: “Spanning-tree of port-vlan <vlan-id> is disabled.”
Bridge identifier	The STP identity of this device.
Active global timers	The global STP timers that are currently active, and their current values. The following timers can be listed: • Hello – The interval between Hello packets. This timer applies only to the root bridge. • Topology Change (TC) – The amount of time during which the topology change flag in Hello packets will be marked, indicating a topology change. This timer applies only to the root bridge. • Topology Change Notification (TCN) – The interval between Topology Change Notification packets sent by a non-root bridge toward the root bridge. This timer applies only to non-root bridges.

Table 9.5: CLI Display of Detailed STP Information for Ports (Continued)

This Field...	Displays...
Port number and STP state	The internal port number and the port's STP state. The internal port number is one of the following: • The port's interface number, if the port is the designated port for the LAN. • The interface number of the designated port from the received BPDU, if the interface is not the designated port for the LAN. The state can be one of the following: • BLOCKING – STP has blocked Layer 2 traffic on this port to prevent a loop. The device or VLAN can reach the root bridge using another port, whose state is FORWARDING. When a port is in this state, the port does not transmit or receive user frames, but the port does continue to receive STP BPDUs. • DISABLED – The port is not participating in STP. This can occur when the port is disconnected or STP is administratively disabled on the port. • FORWARDING – STP is allowing the port to send and receive frames. • LISTENING – STP is responding to a topology change and this port is listening for a BPDU from neighboring bridge(s) in order to determine the new topology. No user frames are transmitted or received during this state. • LEARNING – The port has passed through the LISTENING state and will change to the BLOCKING or FORWARDING state, depending on the results of STP's reconvergence. The port does not transmit or receive user frames during this state. However, the device can learn the MAC addresses of frames that the port receives during this state and make corresponding entries in the MAC table. **Note**: If the state is DISABLED, no further STP information is displayed for the port.
Port Path cost	The port's STP path cost.
Port Priority	This port's STP priority. The value is shown as a hexadecimal number.
Root	The ID assigned by STP to the root bridge for this spanning tree.
Designated Bridge	The MAC address of the designated bridge to which this port is connected. The designated bridge is the device that connects the network segment on the port to the root bridge.
Designated Port	The port number sent from the designated bridge.
Designated Path Cost	The cost to the root bridge as advertised by the designated bridge that is connected to this port. If the bridge is the root bridge itself, then the cost is 0. The identity of the designated bridge is shown in the Designated Bridge field.

Table 9.5: CLI Display of Detailed STP Information for Ports (Continued)

This Field...	Displays...
Active Timers	The current values for the following timers, if active: • Message age – The number of seconds this port has been waiting for a hello message from the root bridge. • Forward delay – The number of seconds that have passed since the last topology change and consequent reconvergence. • Hold time – The number of seconds that have elapsed since transmission of the last Configuration BPDU.
BPDUs Sent and Received	The number of BPDUs sent and received on this port since the software was reloaded.

Displaying Detailed STP Information for a Single Port in a Specific VLAN

Enter a command such as the following to display STP information for an individual port in a specific VLAN.

```
FastIron#show span detail vlan 1 ethernet 7/1
Port 7/1 is FORWARDING
  Port - Path cost: 19, Priority: 128, Root: 0x800000e052a9bb00
  Designated - Bridge: 0x800000e052a9bb00, Interface: 7, Path cost: 0
  Active Timers - None
  BPDUs - Sent: 29, Received: 0
```

Syntax: show span detail [vlan <vlan-id> [ethernet [<slotnum>/]<portnum> | <num>]

Displaying STP State Information for an Individual Interface

To display STP state information for an individual port, you can use the methods in “Displaying STP Information for an Entire Device” on page 9-8 or “Displaying Detailed STP Information for Each Interface” on page 9-12. You also can display STP state information for a specific port using the following method.

To display information for a specific port, enter a command such as the following at any level of the CLI:

```
FastIron#show interface ethernet 3/11

FastEthernet3/11 is up, line protocol is up
  Hardware is FastEthernet, address is 00e0.52a9.bb49 (bia 00e0.52a9.bb49)
  Configured speed auto, actual 100Mbit, configured duplex fdx, actual fdx
  Member of L2 VLAN ID 1, port is untagged, port state is FORWARDING
  STP configured to ON, priority is level0, flow control enabled
  mirror disabled, monitor disabled
  Not member of any active trunks
  Not member of any configured trunks
  No port name
  MTU 1518 bytes, encapsulation ethernet
  5 minute input rate: 352 bits/sec, 0 packets/sec, 0.00% utilization
  5 minute output rate: 0 bits/sec, 0 packets/sec, 0.00% utilization
  1238 packets input, 79232 bytes, 0 no buffer
  Received 686 broadcasts, 0 runts, 0 giants
  0 input errors, 0 CRC, 0 frame, 0 ignored
  529 multicast
  918 packets output, 63766 bytes, 0 underruns
  0 output errors, 0 collisions
```

The STP information is shown in bold type in this example.

Syntax: show interfaces [ethernet [<slotnum>/]<portnum>] | [loopback <num>] | [slot <slot-num>] | [ve <num>] | [brief]

You also can display the STP states of all ports by entering a command such as the following, which uses the **brief** parameter:

```
FastIron#show interface brief

Port  Link State     Dupl Speed Trunk Tag Priori MAC            Name
1/1   Down None      None None  None  No  level0 00e0.52a9.bb00
1/2   Down None      None None  None  No  level0 00e0.52a9.bb01
1/3   Down None      None None  None  No  level0 00e0.52a9.bb02
1/4   Down None      None None  None  No  level0 00e0.52a9.bb03
1/5   Down None      None None  None  No  level0 00e0.52a9.bb04
1/6   Down None      None None  None  No  level0 00e0.52a9.bb05
1/7   Down None      None None  None  No  level0 00e0.52a9.bb06
1/8   Down None      None None  None  No  level0 00e0.52a9.bb07


.
. some rows omitted for brevity
.
3/10  Down None      None None  None  No  level0 00e0.52a9.bb4a
3/11  Up   Forward   Full 100M  None  No  level0 00e0.52a9.bb49
```

In the example above, only one port, 3/11, is forwarding traffic toward the root bridge.

Configuring STP Related Features

STP features extend the operation of standard STP, enabling you to fine tune standard STP and avoid some of its limitations.

This section describes how to configure these parameters on Foundry Layer 3 Switches using the CLI.

Fast Port Span

When STP is running on a device, message forwarding is delayed during the spanning tree recalculation period following a topology change. The STP forward delay parameter specifies the period of time a bridge waits before forwarding data packets. The forward delay controls the listening and learning periods of STP reconvergence. You can configure the forward delay to a value from 4 – 30 seconds. The default is 15 seconds. Thus, using the standard forward delay, convergence requires 30 seconds (15 seconds for listening and an additional 15 seconds for learning) when the default value is used.

This slow convergence is undesirable and unnecessary in some circumstances. The Fast Port Span feature allows certain ports to enter the forwarding state in four seconds. Specifically, Fast Port Span allows faster convergence on ports that are attached to end stations and thus do not present the potential to cause Layer 2 forwarding loops. Because the end stations cannot cause forwarding loops, they can safely go through the STP state changes (blocking to listening to learning to forwarding) more quickly than is allowed by the standard STP convergence time. Fast Port Span performs the convergence on these ports in four seconds (two seconds for listening and two seconds for learning).

In addition, Fast Port Span enhances overall network performance in the following ways:

- Fast Port Span reduces the number of STP topology change notifications on the network. When an end station attached to a Fast Span port comes up or down, the Foundry device does not generate a topology change notification for the port. In this situation, the notification is unnecessary since a change in the state of the host does not affect the network’s topology.

- Fast Port Span eliminates unnecessary MAC cache aging that can be caused by topology change notifications. Bridging devices age out the learned MAC addresses in their MAC caches if the addresses are unrefreshed for a given period of time, sometimes called the MAC aging interval. When STP sends a topology change notification, devices that receive the notification use the value of the STP forward delay to quickly age out their MAC caches. For example, if a device's normal MAC aging interval is 5 minutes, the aging interval changes temporarily to the value of the forward delay (for example, 15 seconds) in response to an STP topology change.

  In normal STP, the accelerated cache aging occurs even when a single host goes up or down. Because Fast Port Span does not send a topology change notification when a host on a Fast Port Span port goes up or down, the unnecessary cache aging that can occur in these circumstances under normal STP is eliminated.

Fast Port Span is a system-wide parameter and is enabled by default. Thus, when you boot a device, all the ports that are attached only to end stations run Fast Port Span. For ports that are not eligible for Fast Port Span, such as ports connected to other networking devices, the device automatically uses the normal STP settings. If a port matches any of the following criteria, the port is ineligible for Fast Port Span and uses normal STP instead:

- The port is 802.1Q tagged
- The port is a member of a trunk group
- The port has learned more than one active MAC address
- An STP Configuration BPDU has been received on the port, thus indicating the presence of another bridge on the port.

You also can explicitly exclude individual ports from Fast Port Span if needed. For example, if the only uplink ports for a wiring closet switch are Gigabit ports, you can exclude the ports from Fast Port Span.

Disabling and Re-enabling Fast Port Span

Fast Port Span is a system-wide parameter and is enabled by default. Thus all ports that are eligible for Fast Port Span use it.

To disable or re-enable Fast Port Span, enter the following commands:

```
FastIron(config)#no fast port-span
FastIron(config)#write memory
```

Syntax: [no] fast port-span

NOTE: The **fast port-span** command has additional parameters that let you exclude specific ports. These parameters are shown in the following section.

To re-enable Fast Port Span, enter the following commands:

```
FastIron(config)#fast port-span
FastIron(config)#write memory
```

Excluding Specific Ports from Fast Port Span

To exclude a port from Fast Port Span while leaving Fast Port Span enabled globally, enter commands such as the following:

```
FastIron(config)#fast port-span exclude ethernet 1
FastIron(config)#write memory
```

To exclude a set of ports from Fast Port Span, enter commands such as the following:

```
FastIron(config)#fast port-span exclude ethernet 1 ethernet 2 ethernet 3
FastIron(config)#write memory
```

To exclude a contiguous (unbroken) range of ports from Fast Span, enter commands such as the following:

```
FastIron(config)#fast port-span exclude ethernet 1 to 24
FastIron(config)#write memory
```

Syntax: [no] fast port-span [exclude ethernet [<slotnum>/]<portnum> [ethernet [<slotnum>/]<portnum> | to [<slotnum>/]<portnum>]]

To re-enable Fast Port Span on a port, enter a command such as the following:

```
FastIron(config)#no fast port-span exclude ethernet 1
FastIron(config)#write memory
```

This command re-enables Fast Port Span on port 1 only and does not re-enable Fast Port Span on other excluded ports. You also can re-enable Fast Port Span on a list or range of ports using the syntax shown above this example.

To re-enable Fast Port Span on all excluded ports, disable and then re-enable Fast Port Span by entering the following commands:

```
FastIron(config)#no fast port-span
FastIron(config)#fast port-span
FastIron(config)#write memory
```

Disabling and then re-enabling Fast Port Span clears the exclude settings and thus enables Fast Port Span on all eligible ports. To make sure Fast Port Span remains enabled on the ports following a system reset, save the configuration changes to the startup-config file after you re-enable Fast Port Span. Otherwise, when the system resets, those ports will again be excluded from Fast Port Span.

802.1W Rapid Spanning Tree (RSTP)

Foundry’s earlier implementation of Rapid Spanning Tree Protocol (RSTP), which was 802.1W Draft 3, provided only a subset of the IEEE 802.1W standard; whereas the 802.1W RSTP feature provides the full standard. The implementation of the 802.1W Draft 3 is referred to as RSTP Draft 3.

RSTP Draft3 will continue to be supported on Foundry devices for backward compatibility. However, customers who are currently using RSTP Draft 3 should migrate to 802.1W.

The 802.1W feature provides rapid traffic reconvergence for point-to-point links within a few milliseconds (0 – 500 milliseconds), following the failure of a bridge or bridge port. This reconvergence occurs more rapidly than the reconvergence provided by the 802.1D (Spanning Tree Protocol (STP)) or by RSTP Draft 3.

NOTE: This rapid convergence will not occur on ports connected to shared media devices, such as hubs. To take advantage of the rapid convergence provided by 802.1W, make sure to explicitly configure all point-to-point links in a topology.

The convergence provided by the standard 802.1W protocol occurs more rapidly than the convergence provided by previous spanning tree protocols because:

- Classic or legacy 802.1D STP protocol requires a newly selected Root port to go through listening and learning stages before traffic convergence can be achieved. The 802.1D traffic convergence time is calculated using the following formula:

  2 x FORWARD_DELAY + BRIDGE_MAX_AGE.

  If default values are used in the parameter configuration, convergence can take up to 50 seconds. (In this document STP will be referred to as 802.1D.)
- RSTP Draft 3 works only on bridges that have Alternate ports, which are the precalculated “next best root port”. (Alternate ports provide back up paths to the root bridge.) Although convergence occurs from 0 – 500 milliseconds in RSTP Draft 3, the spanning tree topology reverts to the 802.1D convergence if an Alternate port is not found.
- Convergence in 802.1w bridge is not based on any timer values. Rather, it is based on the explicit handshakes between Designated ports and their connected Root ports to achieve convergence in less than 500 milliseconds.

Bridges and Bridge Port Roles

A bridge in an 802.1W rapid spanning tree topology is assigned as the root bridge if it has the highest priority (lowest bridge identifier) in the topology. Other bridges are referred to as non-root bridges.

Unique roles are assigned to ports on the root and non-root bridges. Role assignments are based on the following information contained in the Rapid Spanning Tree Bridge Packet Data Unit (RST BPDU):

- Root bridge ID
- Path cost value
- Transmitting bridge ID
- Designated port ID

The 802.1W algorithm uses this information to determine if the RST BPDU received by a port is superior to the RST BPDU that the port transmits. The two values are compared in the order as given above, starting with the Root bridge ID. The RST BPDU with a lower value is considered superior. The superiority and inferiority of the RST BPDU is used to assign a role to a port.

If the value of the received RST BPDU is the same as that of the transmitted RST BPDU, then the port ID in the RST BPDUs are compared. The RST BPDU with the lower port ID is superior. Port roles are then calculated appropriately.

The port's role is included in the BPDU that it transmits. The BPDU transmitted by an 802.1W port is referred to as an RST BPDU, while it is operating in 802.1W mode.

Ports can have one of the following roles:

- Root – Provides the lowest cost path to the root bridge from a specific bridge
- Designated – Provides the lowest cost path to the root bridge from a LAN to which it is connected
- Alternate – Provides an alternate path to the root bridge when the root port goes down
- Backup – Provides a backup to the LAN when the Designated port goes down
- Disabled – Has no role in the topology

Assignment of Port Roles

At system start-up, all 802.1W-enabled bridge ports assume a Designated role. Once start-up is complete, the 802.1W algorithm calculates the superiority or inferiority of the RST BPDU that is received and transmitted on a port.

On a root bridge, each port is assigned a ***Designated port*** role, except for ports on the same bridge that are physically connected together. In these type of ports, the port that receives the superior RST BPDU becomes the ***Backup port***, while the other port becomes the ***Designated port***.

On non-root bridges, ports are assigned as follows:

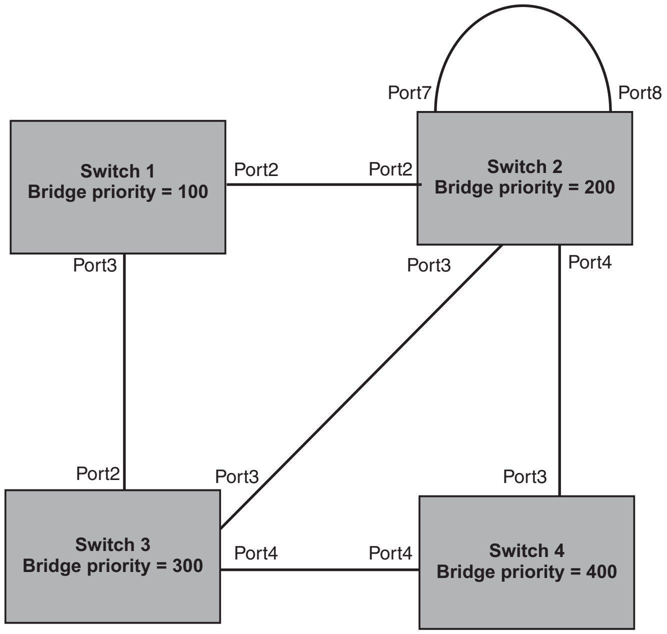
- The port that receives the RST BPDU with the lowest path cost from the root bridge becomes the ***Root port***.
- If two ports on the same bridge are physically connected, the port that receives the superior RST BPDU becomes the ***Backup port***, while the other port becomes the ***Designated port***.
- If a non-root bridge already has a Root port, then the port that receives an RST BPDU that is superior to those it can transmit becomes the ***Alternate port***.
- If the RST BPDU that a port receives is inferior to the RST BPDUs it transmits, then the port becomes a ***Designated port***.
- If the port is down or if 802.1W is disabled on the port, that port is given the role of ***Disabled port***. Disabled ports have no role in the topology. However, if 802.1W is enabled on a port with a link down and the link of that port comes up, then that port assumes one of the following port roles: Root, Designated, Alternate, or Backup.

The following example (Figure 9.1) explains role assignments in a simple RSTP topology.

NOTE: All examples in this document assume that all ports in the illustrated topologies are point-to-point links and are homogeneous (they have the same path cost value) unless otherwise specified.

The topology in Figure 9.1 contains four bridges. Switch 1 is the root bridge since it has the lowest bridge priority. Switch 2 through Switch 4 are non-root bridges.

Figure 9.1 Simple 802.1W Topology



Ports on Switch 1

All ports on Switch 1, the root bridge, are assigned Designated port roles.

Ports on Switch 2

Port2 on Switch 2 directly connects to the root bridge; therefore, Port2 is the Root port.

Switch 2's bridge priority value is superior to that of Switch 3 and Switch 4; therefore, the ports on Switch 2 that connect to Switch 3 and Switch 4 are given the Designated port role.

Furthermore, Port7 and Port8 on Switch 2 are physically connected. The RST BPDUs transmitted by Port7 are superior to those Port8 transmits. Therefore, Port8 is the Backup port and Port7 is the Designated port.

Ports on Switch 3

Port2 on Switch 3 directly connects to the Designated port on the root bridge; therefore, it assumes the Root port role.

The root path cost of the RST BPDUs received on Port4/Switch 3 is inferior to the RST BPDUs transmitted by the port; therefore, Port4/Switch 3 becomes the Designated port.

Similarly Switch 3 has a bridge priority value inferior to Switch 2. Port3 on Switch 3 connects to Port 3 on Switch 2. This port will be given the Alternate port role, since a Root port is already established on this bridge.

Ports Switch 4

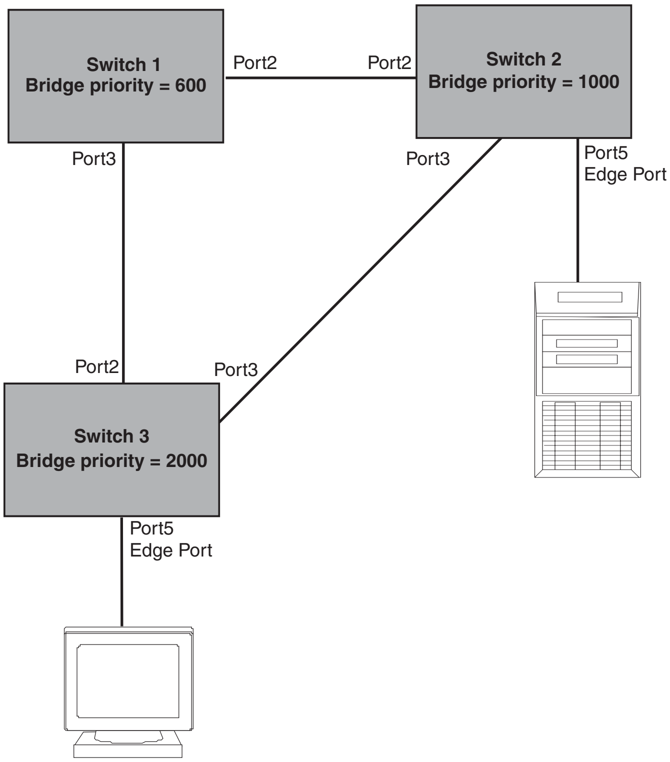
Switch 4 is not directly connected to the root bridge. It has two ports with superior incoming RST BPDUs from two separate LANs: Port3 and Port4. The RST BPDUs received on Port3 are superior to the RST BPDUs received on port 4; therefore, Port3 becomes the Root port and Port4 becomes the Alternate port.

Edge Ports and Edge Port Roles

Foundry's implementation of 802.1W allows ports that are configured as Edge ports to be present in an 802.1W topology. (Figure 9.2). Edge ports are ports of a bridge that connect to workstations or computers. Edge ports do not register any incoming BPDU activities.

Edge ports assume Designated port roles. Port flapping does not cause any topology change events on Edge ports since 802.1W does not consider Edge ports in the spanning tree calculations.

Figure 9.2 Topology with Edge Ports



However, if any incoming RST BPDU is received from a previously configured Edge port, 802.1W automatically makes the port as a non-edge port. This is extremely important to ensure a loop free Layer 2 operation since a non-edge port is part of the active RSTP topology.

The 802.1W protocol can auto-detect an Edge port and a non-edge port. An administrator can also configure a port to be an Edge port using the CLI. It is recommended that Edge ports are configured explicitly to take advantage of the Edge port feature, instead of allowing the protocol to auto-detect them.

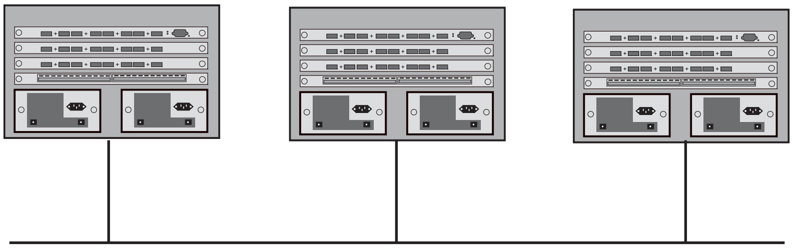
Point-to-Point Ports

To take advantage of the 802.1W features, ports on an 802.1W topology should be explicitly configured as point-to-point links using the CLI. Shared media should not be configured as point-to-point links.

NOTE: Configuring shared media or non-point-to-point links as point-to-point links could lead to Layer 2 loops.

The topology in Figure 9.3 is an example of shared media that should not be configured as point-to-point links. In Figure 9.3, a port on a bridge communicates or is connected to at least two ports.

Figure 9.3 Example of Shared Media



Bridge Port States

Ports roles can have one of the following states:

- Forwarding – 802.1W is allowing the port to send and receive all packets.
- Discarding – 802.1W has blocked data traffic on this port to prevent a loop. The device or VLAN can reach the root bridge using another port, whose state is forwarding. When a port is in this state, the port does not transmit or receive data frames, but the port does continue to receive RST BPDUs. This state corresponds to the listening and blocking states of 802.1D.
- Learning – 802.1W is allowing MAC entries to be added to the filtering database but does not permit forwarding of data frames. The device can learn the MAC addresses of frames that the port receives during this state and make corresponding entries in the MAC table.
- Disabled – The port is not participating in 802.1W. This can occur when the port is disconnected or 802.1W is administratively disabled on the port.

A port on a non-root bridge with the role of Root port is always in a forwarding state. If another port on that bridge assumes the Root port role, then the old Root port moves into a discarding state as it assumes another port role.

A port on a non-root bridge with a Designated role starts in the discarding state. When that port becomes elected to the Root port role, 802.1W quickly places it into a forwarding state. However, if the Designated port is an Edge port, then the port starts and stays in a forwarding state and it cannot be elected as a Root port.

A port with an Alternate or Backup role is always in a discarding state. If the port's role changes to Designated, then the port changes into a forwarding state.

If a port on one bridge has a Designated role and that port is connected to a port on another bridge that has an Alternate or Backup role, the port with a Designated role cannot be given a Root port role until two instances of the forward delay timer expires on that port.

Edge Port and Non-Edge Port States

As soon as a port is configured as an Edge port using the CLI, it goes into a forwarding state instantly (in less than 100 msec):

When the link to a port comes up and 802.1W detects that the port is an Edge port, that port instantly goes into a forwarding state.

If 802.1W detects that port as a non-edge port, the port state is changed as determined by the result of processing the received RST BPDU. The port state change occurs within four seconds of link up or after two hello timer expires on the port.

Changes to Port Roles and States

To achieve convergence in a topology, a port's role and state changes as it receives and transmits new RST BPDUs. Changes in a port's role and state constitute a topology change. Besides the superiority and inferiority of the RST BPDU, bridge-wide and per-port state machines are used to determine a port's role as well as a port's state. Port state machines also determine when port role and state changes occur.

State Machines

The bridge uses the Port Role Selection state machine to determine if port role changes are required on the bridge. This state machine performs a computation when one of the following events occur:

- New information is received on any port on the bridge
- The timer expires for the current information on a port on the bridge

Each port uses the following state machines:

- Port Information – This state machine keeps track of spanning-tree information currently used by the port. It records the origin of the information and ages out any information that was derived from an incoming BPDU.
- Port Role Transition – This state machine keeps track of the current port role and transitions the port to the appropriate role when required. It moves the Root port and the Designated port into forwarding states and moves the Alternate and Backup ports into discarding states.
- Port Transmit – This state machine is responsible for BPDU transmission. It checks to ensure only the maximum number of BPDUs per hello interval are sent every second. Based on what mode it is operating in, it sends out either legacy BPDUs or RST BPDUs. In this document legacy BPDUs are also referred to as STP BPDUs.
- Port Protocol Migration – This state machine deals with compatibility with 802.1D bridges. When a legacy BPDU is detected on a port, this state machine configures the port to transmit and receive legacy BPDUs and operate in the legacy mode.
- Topology Change – This state machine detects, generates, and propagates topology change notifications. It acknowledges Topology Change Notice (TCN) messages when operating in 802.1D mode. It also flushes the MAC table when a topology change event takes place.
- Port State Transition – This state machine transitions the port to a discarding, learning, or forwarding state and performs any necessary processing associated with the state changes.
- Port Timers – This state machine is responsible for triggering any of the state machines described above, based on expiration of specific port timers.

In contrast to the 802.1D standard, the 802.1W standard does not have any bridge specific timers. All timers in the CLI are applied on a per-port basis, even though they are configured under bridge parameters.

802.1W state machines attempt to quickly place the ports into either a forwarding or discarding state. Root ports are quickly placed in forwarding state when both of the following events occur:

- It is assigned to be the Root port.
- It receives an RST BPDU with a proposal flag from a Designated port. The proposal flag is sent by ports with a Designated role when they are ready to move into a forwarding state.

When a the role of Root port is given to another port, the old Root port is instructed to reroot. The old Root port goes into a discarding state and negotiates with its peer port for a new role and a new state. A peer port is the port on the other bridge to which the port is connected. For example, in Figure 9.4, Port1 of Switch 200 is the peer port of Port2 of Switch 100.

A port with a Designated role is quickly placed into a forwarding state if one of the following occurs:

- The Designated port receives an RST BPDU that contains an agreement flag from a Root port
- The Designated port is an Edge port

However, a Designated port that is attached to an Alternate port or a Backup port must wait until the forward delay timer expires twice on that port while it is still in a Designated role, before it can proceed to the forwarding state.

Backup ports are quickly placed into discarding states.

Alternate ports are quickly placed into discarding states.

A port operating in 802.1W mode may enter a learning state to allow MAC entries to be added to the filtering database; however, this state is transient and lasts only a few milliseconds, if the port is operating in 802.1W mode and if the port meets the conditions for rapid transition.

Handshake Mechanisms

To rapidly transition a Designated or Root port into a forwarding state, the Port Role Transition state machine uses handshake mechanisms to ensure loop free operations. It uses one type of handshake if no Root port has been assigned on a bridge, and another type if a Root port has already been assigned.

Handshake When No Root Port is Elected

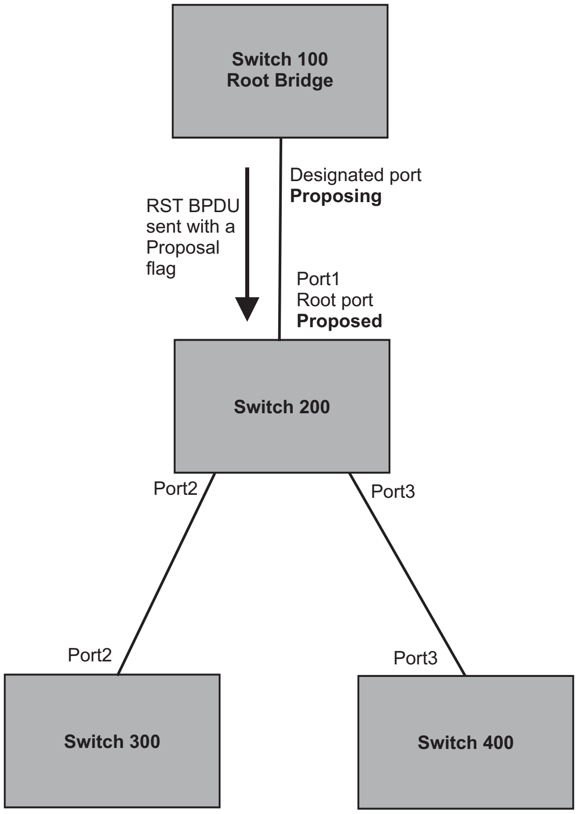
If a Root port has not been assigned on a bridge, 802.1W uses the Proposing -> Proposed -> Sync -> Synced -> Agreed handshake:

- Proposing – The Designated port on the root bridge sends an RST BPDU packet to its peer port that contains a proposal flag. The proposal flag is a signal that indicates that the Designated port is ready to put itself in a forwarding state (Figure 9.4). The Designated port continues to send this flag in its RST BPDU until it is placed in a forwarding state (Figure 9.7) or is forced to operate in 802.1D mode. (See “Compatibility of 802.1W with 802.1D” on page 43.)
- Proposed – When a port receives an RST BPDU with a proposal flag from the Designated port on its point-to-point link, it asserts the Proposed signal and one of the following occurs (Figure 9.4):
  - If the RST BPDU that the port receives is superior to what it can transmit, the port assumes the role of a Root port. (See the section on “Bridges and Bridge Port Roles” on page 9-19.)
  - If the RST BPDU that the port receives is inferior to what it can transmit, then the port is given the role of Designated port.

NOTE: Proposed will never be asserted if the port is connected on a shared media link.

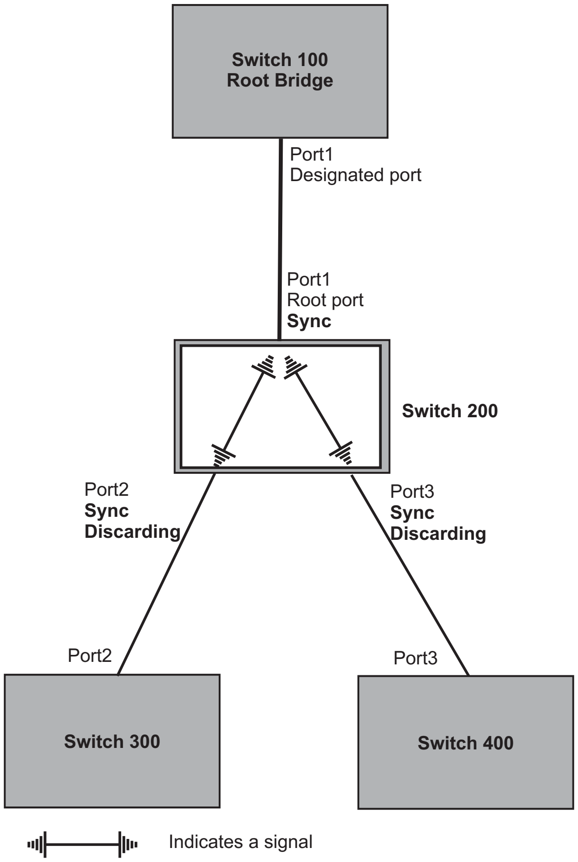
In Figure 9.4, Port3/Switch 200 is elected as the Root port

Figure 9.4 Proposing and Proposed Stage



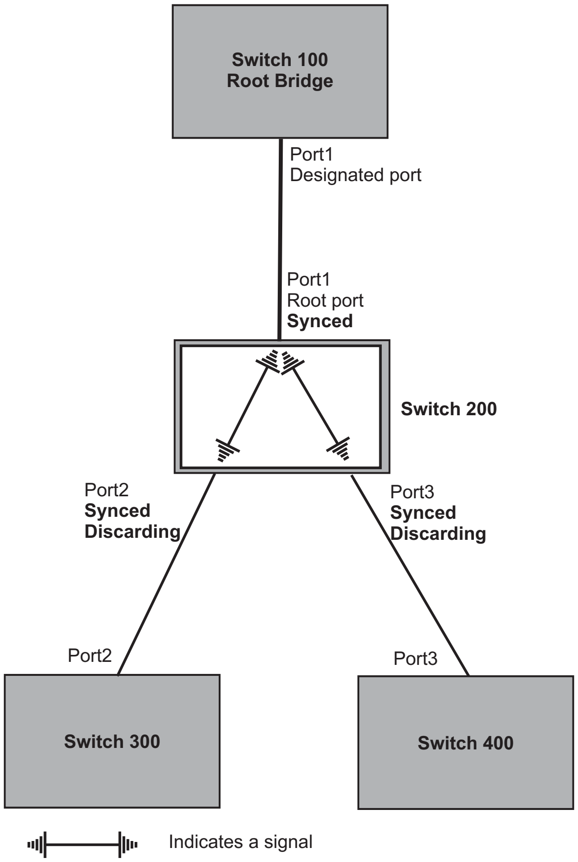
- Sync – Once the Root port is elected, it sets a sync signal on all the ports on the bridge. The signal tells the ports to synchronize their roles and states (Figure 9.5). Ports that are non-edge ports with a role of Designated port change into a discarding state. These ports have to negotiate with their peer ports to establish their new roles and states.

Figure 9.5 Sync Stage



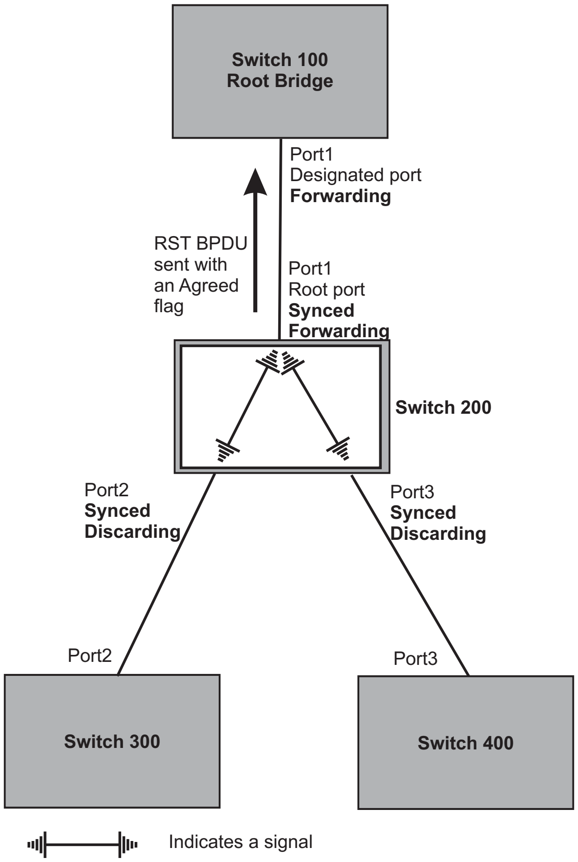
- Synced – Once the Designated port changes into a discarding state, it asserts a synced signal. Immediately, Alternate ports and Backup ports are synced. The Root port monitors the synced signals from all the bridge ports. Once all bridge ports asserts a synced signal, the Root port asserts its own synced signal (Figure 9.6).

Figure 9.6 Synced Stage



- Agreed – The Root port sends back an RST BPDU containing an agreed flag to its peer Designated port and moves into the forwarding state. When the peer Designated port receives the RST BPDU, it rapidly transitions into a forwarding state.

Figure 9.7 Agree Stage



At this point, the handshake mechanism is complete between Switch 100, the root bridge, and Switch 200.

Switch 200 updates the information on the Switch 200's Designated ports (Port2 and Port3) and identifies the new root bridge. The Designated ports send RST BPDUs, containing proposal flags, to their downstream bridges, without waiting for the hello timers to expire on them. This process starts the handshake with the downstream bridges.

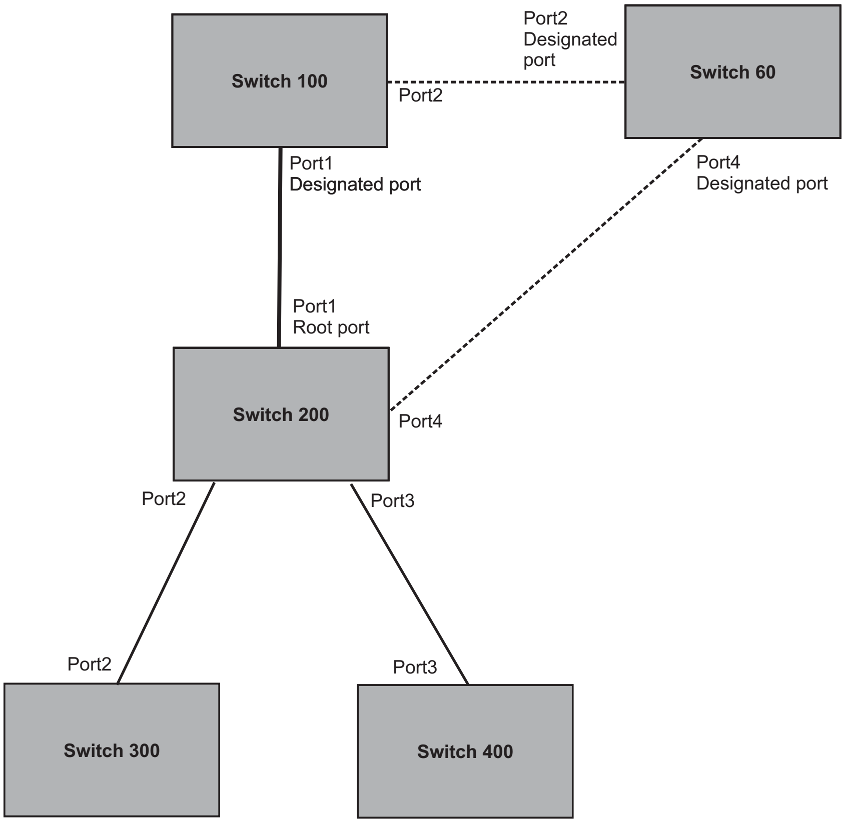
For example, Port2/Switch 200 sends an RST BPDU to Port2/Switch 300 that contains a proposal flag. Port2/ Switch 300 asserts a proposed signal. Ports in Switch 300 then set sync signals on the ports to synchronize and negotiate their roles and states. Then the ports assert a synced signal and when the Root port in Switch 300 asserts it's synced signal, it sends an RST BPDU to Switch 200 with an agreed flag.

This handshake is repeated between Switch 200 and Switch 400 until all Designated and Root ports are in forwarding states.

Handshake When a Root Port Has Been Elected

If a non-root bridge already has a Root port, 802.1W uses a different type of handshake. For example, in Figure 9.8, a new root bridge is added to the topology.

Figure 9.8 Addition of a New Root Bridge

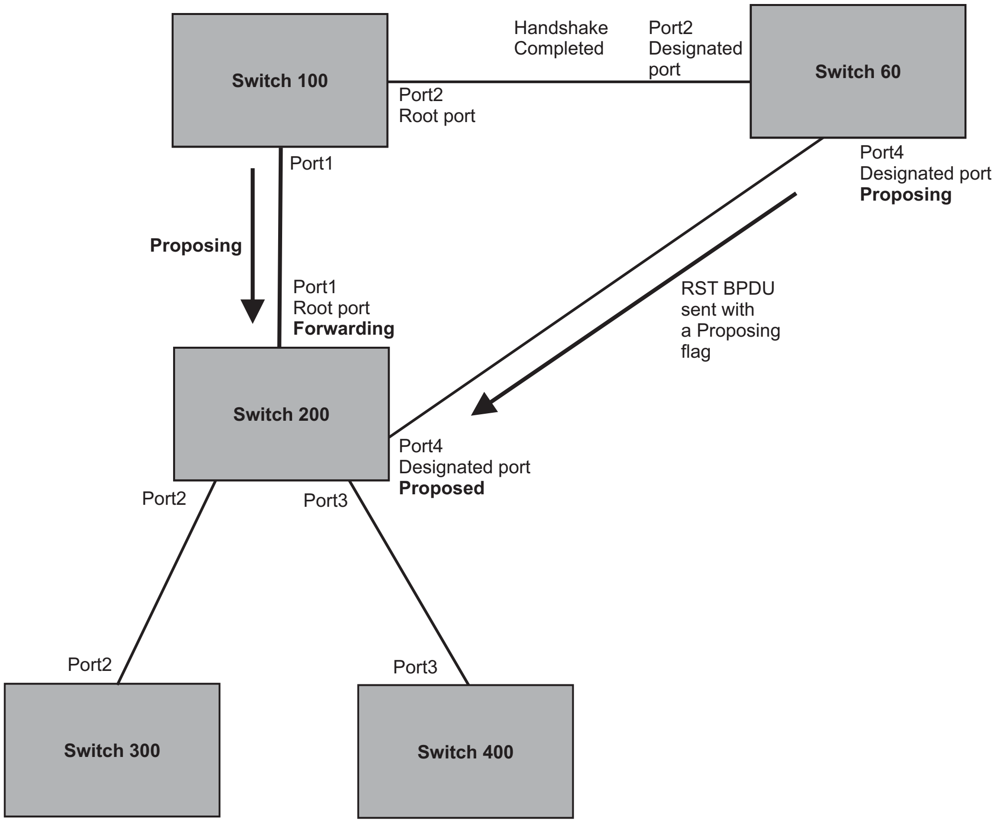


The handshake that occurs between Switch 60 and Switch 100 follows the one described in the previous section ("Handshake When No Root Port is Elected" on page 9-24). The former root bridge becomes a non-root bridge and establishes a Root port (Figure 9.9).

However, since Switch 200 already had a Root port in a forwarding state, 802.1W uses the Proposing -> Proposed -> Sync and Reroot -> Sync and Rerooted -> Rerooted and Synced -> Agreed handshake:

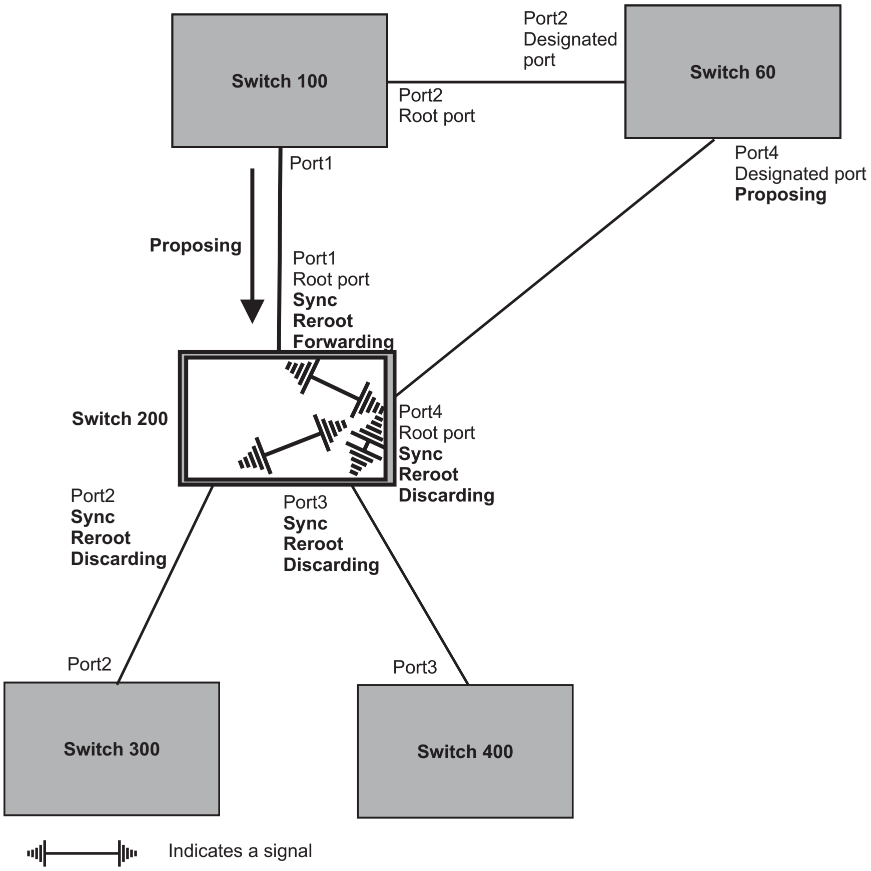
- Proposing and Proposed – The Designated port on the new root bridge (Port4/Switch 60) sends an RST BPDU that contains a proposing signal to Port4/Switch 200 to inform the port that it is ready to put itself in a forwarding state (Figure 9.9). 802.1W algorithm determines that the RST BPDU that Port4/Switch 200 received is superior to what it can generate, so Port4/Switch 200 assumes a Root port role.

Figure 9.9 New Root Bridge Sending a Proposal Flag



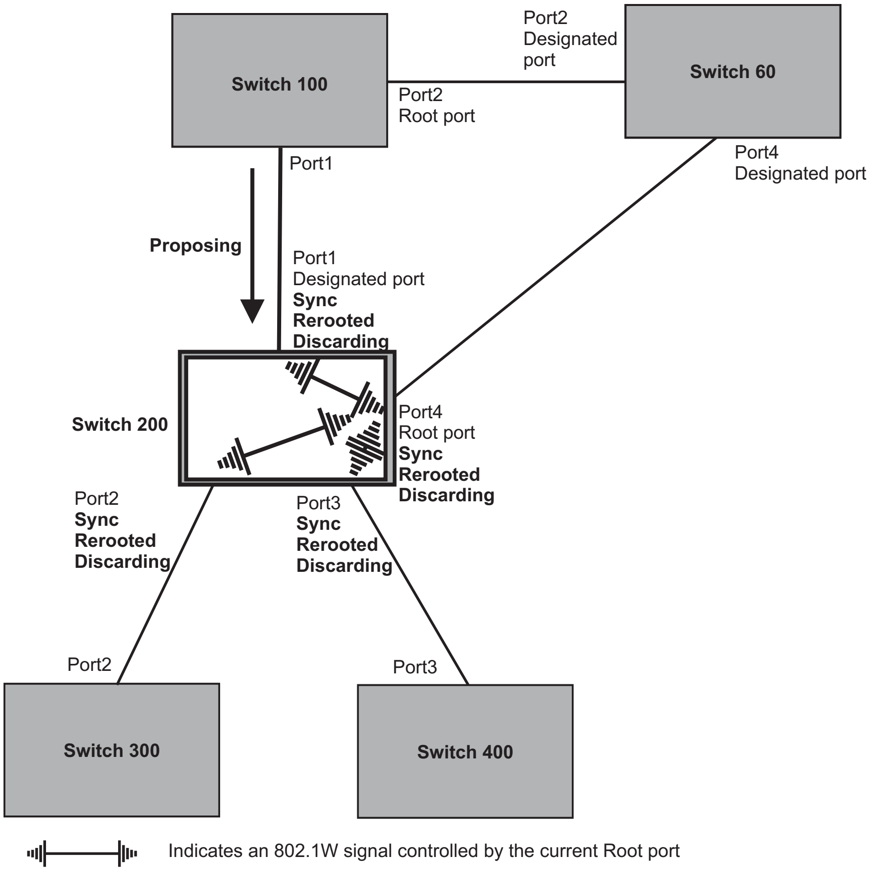
- Sync and Reroot – The Root port then asserts a sync and a reroot signal on all the ports on the bridge. The signal tells the ports that a new Root port has been assigned and they are to renegotiate their new roles and states. The other ports on the bridge assert their sync and reroot signals. Information about the old Root port is discarded from all ports. Designated ports change into discarding states (Figure 9.10).

Figure 9.10 Sync and Reroot



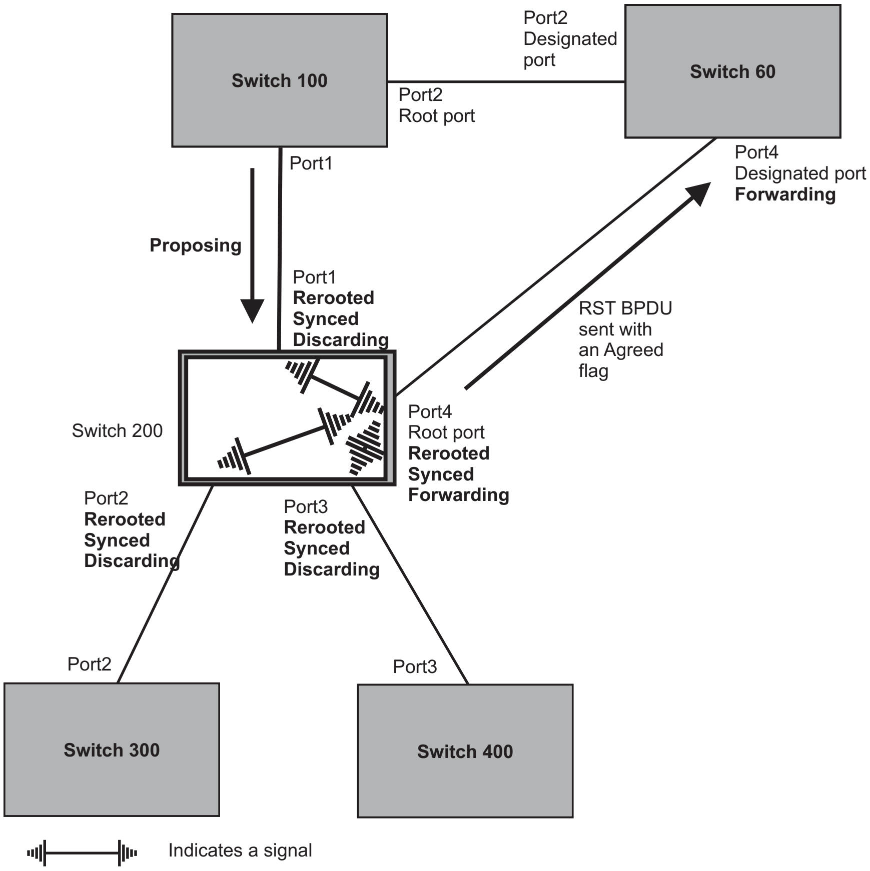
- Sync and Rerooted – When the ports on Switch 200 have completed the reroot phase, they assert their rerooted signals and continue to assert their sync signals as they continue in their discarding states. They also continue to negotiate their roles and states with their peer ports (Figure 9.11).

Figure 9.11 Sync and Rerooted



- Synced and Agree – When all the ports on the bridge assert their synced signals, the new Root port asserts its own synced signal and sends an RST BPDU to Port4/Switch 60 that contains an agreed flag (Figure 9.11). The Root port also moves into a forwarding state.

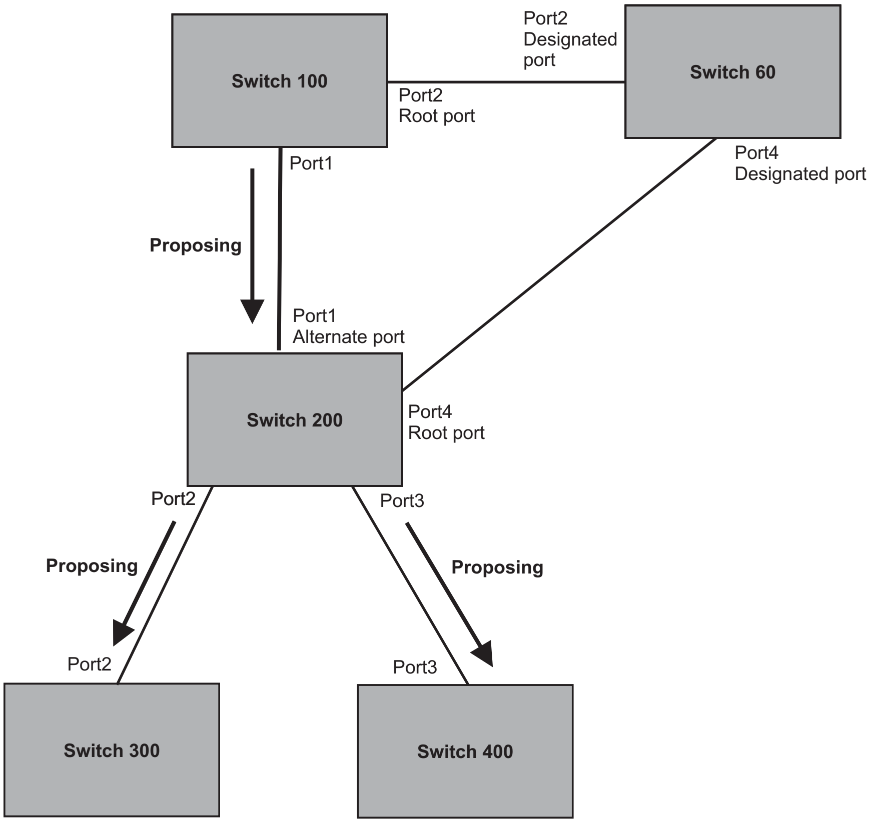
Figure 9.12 Rerooted, Synced, and Agreed



The old Root port on Switch 200 becomes an Alternate Port (Figure 9.13). Other ports on that bridge are elected to appropriate roles.

The Designated port on Switch 60 goes into a forwarding state once it receives the RST BPDU with the agreed flag.

Figure 9.13 Handshake Completed After Election of New Root Port



Recall that Switch 200 sent the agreed flag to Port4/Switch 60 and not to Port1/Switch 100 (the port that connects Switch 100 to Switch 200). Therefore, Port1/Switch 100 does not go into forwarding state instantly. It waits until two instances of the forward delay timer expires on the port before it goes into forwarding state.

At this point the handshake between the Switch 60 and Switch 200 is complete.

The remaining bridges (Switch 300 and Switch 400) may have to go through the reroot handshake if a new Root port needs to be assigned.

Convergence in a Simple Topology

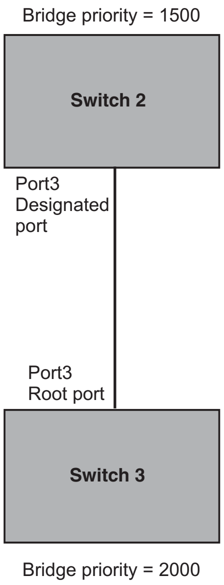
The examples in this section illustrate how 802.1W convergence occurs in a simple Layer 2 topology at start-up.

NOTE: The remaining examples assume that the appropriate handshake mechanisms occur as port roles and states change.

Convergence at Start Up

In Figure 9.14, two bridges Switch 2 and Switch 3 are powered up. There are point-to-point connections between Port3/Switch 2 and Port3/Switch 3.

Figure 9.14 Convergence Between Two Bridges



At power up, all ports on Switch 2 and Switch 3 assume Designated port roles and are at discarding states before they receive any RST BPDU.

Port3/Switch 2, with a Designated role, transmits an RST BPDU with a proposal flag to Port3/Switch 3. A ports with a Designated role sends the proposal flag in its RST BPDU when they are ready to move to a forwarding state.

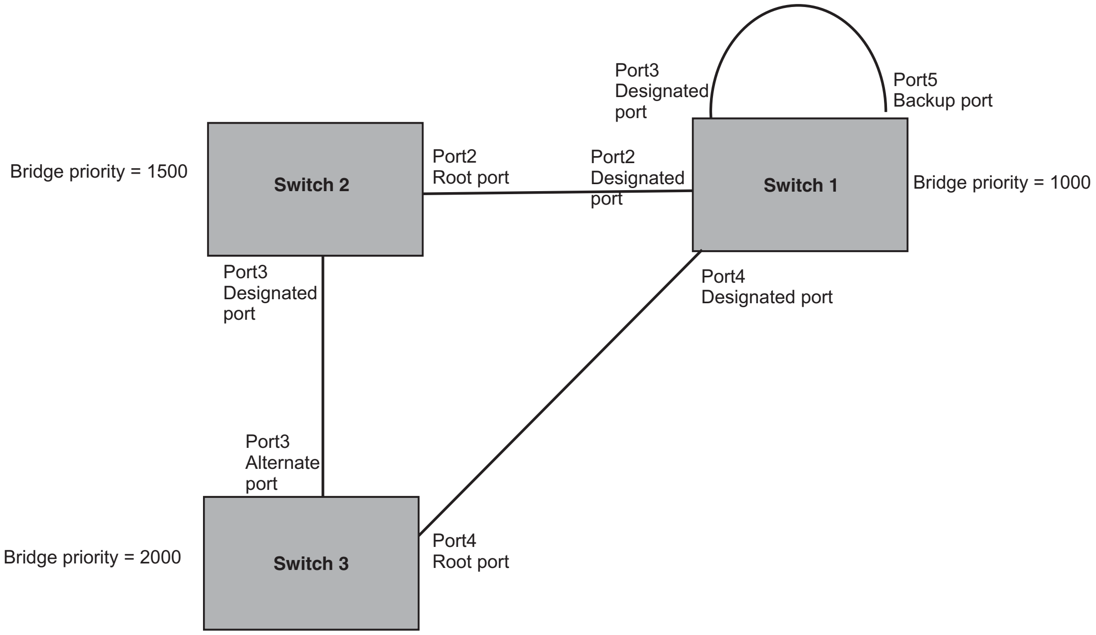
Port3/Switch 3, which starts with a role of Designated port, receives the RST BPDU and finds that it is superior to what it can transmit; therefore, Port3/Switch 3 assumes a new port role, that of a Root port. Port3/Switch 3 transmits an RST BPDU with an agreed flag back to Switch 2 and immediately goes into a forwarding state.

Port3/Switch 2 receives the RST BPDU from Port3/Switch 3 and immediately goes into a forwarding state.

Now 802.1W has fully converged between the two bridges, with Port3/Switch 3 as an operational root port in forwarding state and Port3/Switch 2 as an operational Designated port in forwarding state.

Next, Switch 1 is powered up (Figure 9.15).

Figure 9.15 Simple Layer 2 Topology



The point-to-point connections between the three bridges are as follows:

- Port2/Switch 1 and Port2/Switch 2
- Port4/Switch 1 and Port4/Switch 3
- Port3/Switch 2 and Port3/Switch 3

Ports 3 and 5 on Switch 1 are physically connected together.

At start up, the ports on Switch 1 assume Designated port roles, which are in discarding state. They begin sending RST BPDUs with proposal flags to move into a forwarding state.

When Port4/Switch 3 receives these RST BPDUs 802.1W algorithm determines that they are better than the RST BPDUs that were previously received on Port3/Switch 3. Port4/Switch 3 is now selected as Root port. This new assignment signals Port3/Switch 3 to begin entering the discarding state and to assume an Alternate port role. As it goes through the transition, Port3/Switch 3 negotiates a new role and state with its peer port, Port3/Switch 2.

Port4/Switch 3 sends an RST BPDU with an agreed flag to Port4/Switch 1. Both ports go into forwarding states.

Port2/Switch 2 receives an RST BPDU. The 802.1W algorithm determines that these RST BPDUs that are superior to any that any port on Switch 2 can transmit; therefore, Port2/Switch 2 assumes the role of a Root port.

The new Root port then signals all ports on the bridge to start synchronization. Since none of the ports are Edge ports, they all enter the discarding state and assume the role of Designated ports. Port3/Switch 2, which previously had a Designated role with a forwarding state, starts the discarding state. They also negotiate port roles and states with their peer ports. Port3/Switch 2 also sends an RST BPU to Port3/Switch 3 with a proposal flag to request permission go into a forwarding state.

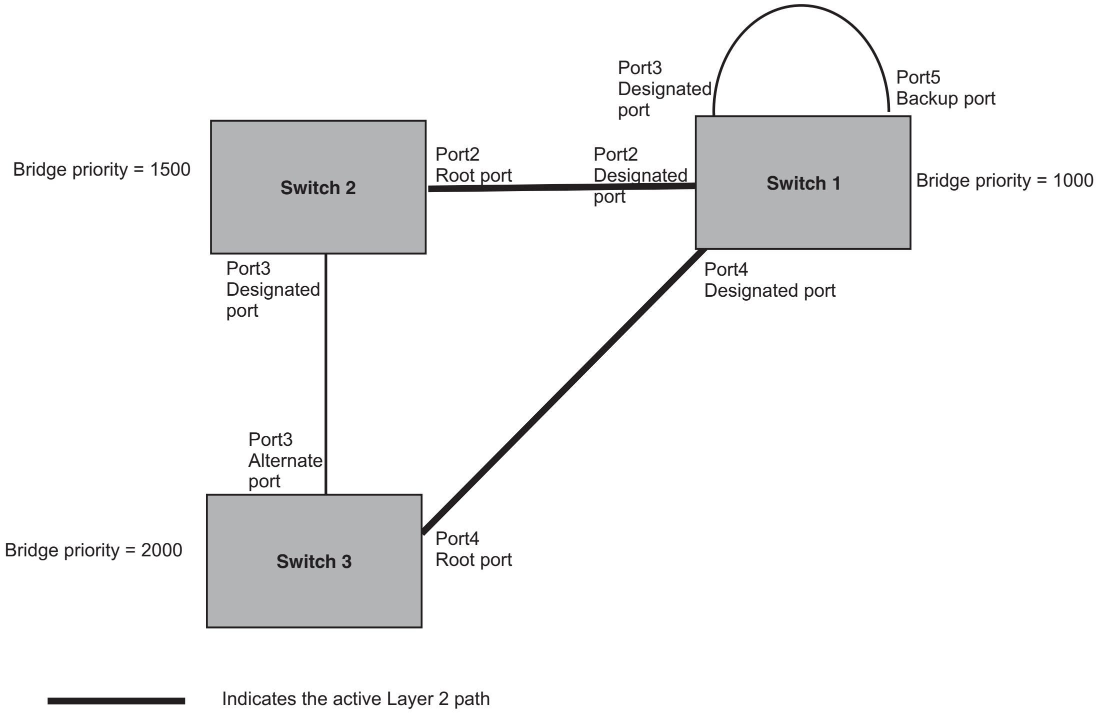
The Port2/Switch 2 bridge also sends an RST BPDU with an agreed flag Port2/Switch 1 that Port2 is the new Root port. Both ports go into forwarding states.

Now, Port3/Switch 3 is currently in a discarding state and is negotiating a port role. It received RST BPDUs from Port3/Switch 2. The 802.1W algorithm determines that the RST BPDUs Port3/Switch 3 received are superior to those it can transmit; however, they are not superior to those that are currently being received by the current Root port (Port4). Therefore, Port3 retains the role of Alternate port.

Ports 3/Switch 1 and Port5/Switch 1 are physically connected. Port5/Switch 1 received RST BPDUs that are superior to those received on Port3/Switch 1; therefore, Port5/Switch 1 is given the Backup port role while Port3 is given the Designated port role. Port3/Switch 1, does not go directly into a forwarding state. It waits until the forward delay time expires twice on that port before it can proceed to the forwarding state.

Once convergence is achieved, the active Layer 2 forwarding path converges as shown in Figure 9.16.

Figure 9.16 Active Layer 2 Path

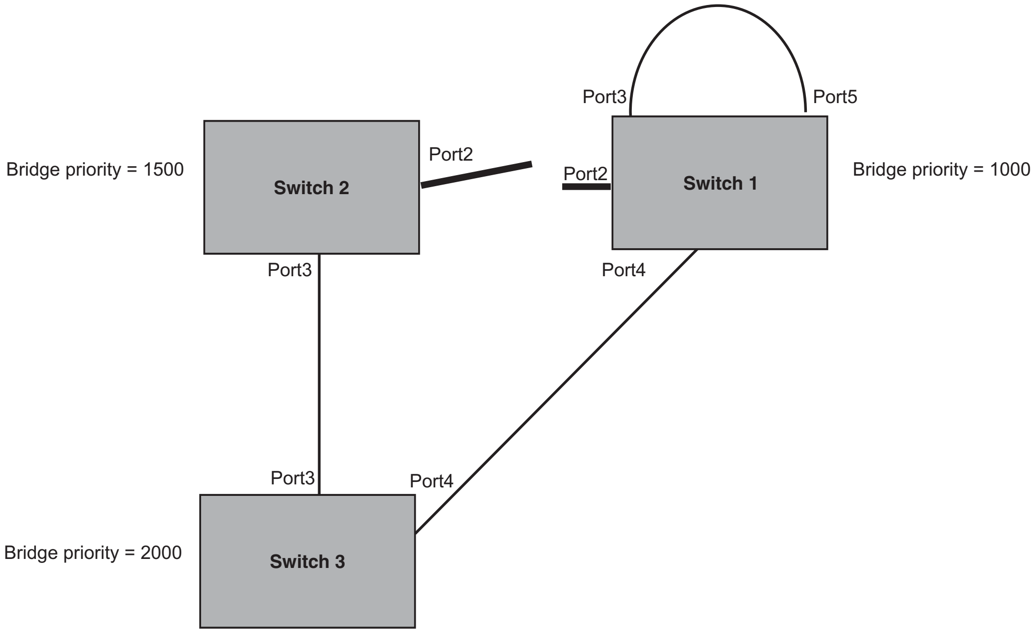


Convergence After a Link Failure

What happens if a link in the 802.1W topology fails?

For example, Port2/Switch, which is the port that connects Switch 2 to the root bridge (Switch 1), fails. Both Switch 2 and Switch 1 notice the topology change (Figure 9.17).

Figure 9.17 Link Failure in the Topology



Switch 1 sets its Port2 into a discarding state.

At the same time, Switch 2 assumes the role of a root bridge since its root port failed and it has no operational Alternate port. Port3/Switch 2, which currently has a Designated port role, sends an RST BPDU to Switch 3. The RST BPDU contains a proposal flag and a bridge ID of Switch 2 as its root bridge ID.

When Port3/Switch 3 receives the RST BPDUs, 802.1W algorithm determines that they are inferior to those that the port can transmit. Therefore, Port3/Switch 3 is given a new role, that of a Designated port. Port3/Switch 3 then sends an RST BPDU with a proposal flag to Switch 2, along with the new role information. However, the root bridge ID transmitted in the RST BPDU is still Switch 1.

When Port3/Switch 2 receives the RST BPDU, 802.1W algorithm determines that it is superior to the RST BPDU that it can transmit; therefore, Port3/Switch 2 receives a new role; that of a Root port. Port3/Switch 2 then sends an RST BPDU with an agreed flag to Port3/Switch 3. Port3/Switch 2 goes into a forwarding state.

When Port3/Switch 3 receives the RST BPDU that Port3/Switch 2 sent, Port3/Switch 3 changes into a forwarding state, which then completes the full convergence of the topology.

Convergence at Link Restoration

When Port2/Switch 2 is restored, both Switch 2 and Switch 1 recognize the change. Port2/Switch 1 starts assuming the role of a Designated port and sends an RST BPDU containing a proposal flag to Port2/Switch 2.

When Port2/Switch 2 receives the RST BPDUs, 802.1W algorithm determines that the RST BPDUs the port received are better than those received on Port3/Switch 3; therefore, Port2/Switch 2 is given the role of a Root port. All the ports on Switch 2 are informed that a new Root port has been assigned which then signals all the ports to synchronize their roles and states. Port3/Switch 2, which was the previous Root port, enters a discarding state and negotiates with other ports on the bridge to establish its new role and state, until it finally assumes the role of a Designated port.

Next, the following happens:

- Port3/Switch 2, the Designated port, sends an RST BPDU, with a proposal flag to Port3/Switch 3.
- Port2/Switch 2 also sends an RST BPDU with an agreed flag to Port2/Switch 1 and then places itself into a forwarding state.

When Port2/Switch 1 receives the RST BPDU with an agreed flag sent by Port2/Switch 2, it puts that port into a forwarding state. The topology is now fully converged.

When Port3/Switch 3 receives the RST BPDU that Port3/Switch 2 sent, 802.1W algorithm determines that these RST BPDUs are superior to those that Port3/Switch 3 can transmit. Therefore, Port3/Switch 3 is given a new role, that of an Alternate port. Port3/Switch 3 immediately enters a discarding state.

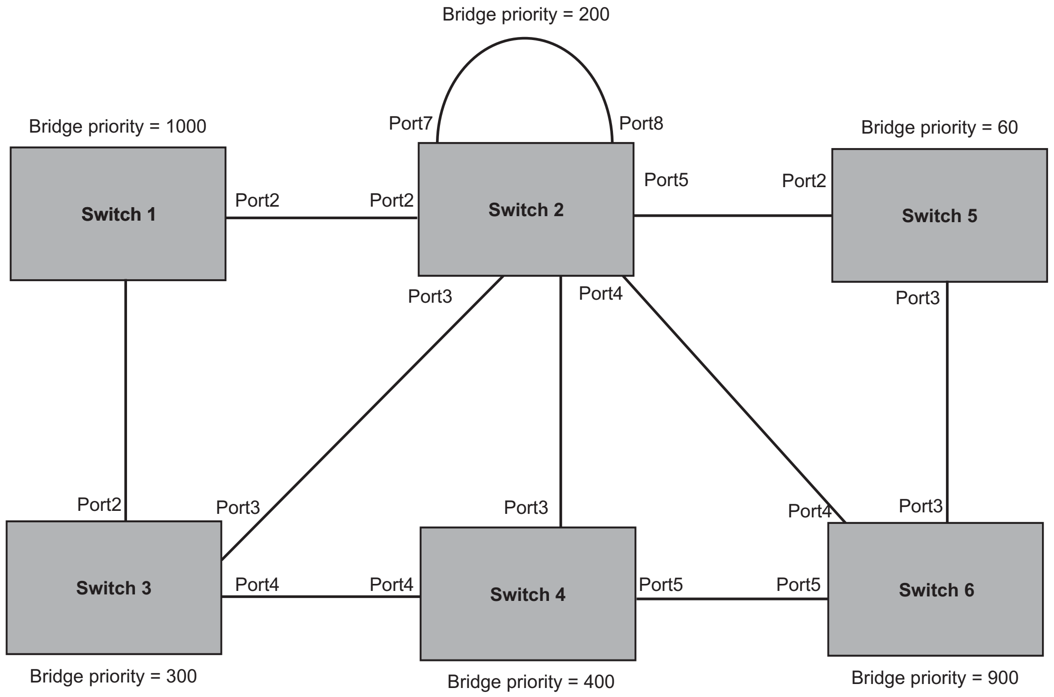
Now Port3/Switch 2 does not go into a forwarding state instantly like the Root port. It waits until the forward delay timer expires twice on that port while it is still in a Designated role, before it can proceed to the forwarding state. The wait, however, does not cause a denial of service, since the essential connectivity in the topology has already been established.

When fully restored, the topology is the same as that shown on Figure 9.15.

Convergence in a Complex 802.1W Topology

The following is an example of a complex 802.1W topology.

Figure 9.18 Complex 802.1W Topology



In Figure 9.18, Switch 5 is selected as the root bridge since it is the bridge with the highest priority. Lines in the figure show the point-to-point connection to the bridges in the topology.

Switch 5 sends an RST BPDU that contains a proposal flag to Port5/Switch 2. When handshakes are completed in Switch 5, Port5/Switch 2 is selected as the Root port on Switch 2. All other ports on Switch 2 are given Designated port role with discarding states.

Port5/Switch 2 then sends an RST BPDU with an agreed flag to Switch 5 to confirm that it is the new Root port and the port enters a forwarding state. Port7 and Port8 are informed of the identity of the new Root port. 802.1W algorithm selects Port7 as the Designated port while Port8 becomes the Backup port.

Port3/Switch 5 sends an RST BPDU to Port3/Switch 6 with a proposal flag. When Port3/Switch 5 receives the RST BPDU, handshake mechanisms select Port3 as the Root port of Switch 6. All other ports are given a Designated port role with discarding states. Port3/Switch 6 then sends an RST BPDU with an agreed flag to Port3/Switch 5 to confirm that it is the Root port. The Root port then goes into a forwarding state.

Now, Port4/Switch 6 receives RST BPDUs that are superior to what it can transmit; therefore, it is given the Alternate port role. The port remains in discarding state.

Port5/Switch 6 receives RST BPDUs that are inferior to what it can transmit. The port is then given a Designated port role.

Next Switch 2 sends RST BPDUs with a proposal flag to Port3/Switch 4. Port3 becomes the Root port for the bridge; all other ports are given a Designated port role with discarding states. Port3/Switch 4 sends an RST BPDU with an agreed flag to Switch 2 to confirm that it is the new Root port. The port then goes into a forwarding state.

Now Port4/Switch 4 receives an RST BPDU that is superior to what it can transmit. The port is then given an Alternate port role, and remains in discarding state.

Likewise, Port5/Switch 4 receives an RST BPDU that is superior to what it can transmit. The port is also given an Alternate port role, and remains in discarding state.

Port2/Switch 2 transmits an RST BPDU with a proposal flag to Port2/Switch 1. Port2/Switch 1 becomes the Root port. All other ports on Switch 1 are given Designated port roles with discarding states.

Port2/Switch 1 sends an RST BPDU with an agreed flag to Port2/Switch 2 and Port2/Switch 1 goes into a forwarding state.

Port3/Switch 1 receives an RST BPDUs that is inferior to what it can transmit; therefore, the port retains its Designated port role and goes into forwarding state only after the forward delay timer expires twice on that port while it is still in a Designated role.

Port3/Switch 2 sends an RST BPDU to Port3/Switch 3 that contains a proposal flag. Port3/Switch 3 becomes the Root port, while all other ports on Switch 3 are given Designated port roles and go into discarding states. Port3/Switch 3 sends an RST BPDU with an agreed flag to Port3/Switch 2 and Port3/Switch 3 goes into a forwarding state.

Now, Port2/Switch 3 receives an RST BPDUs that is superior to what it can transmit so that port is given an Alternate port state.

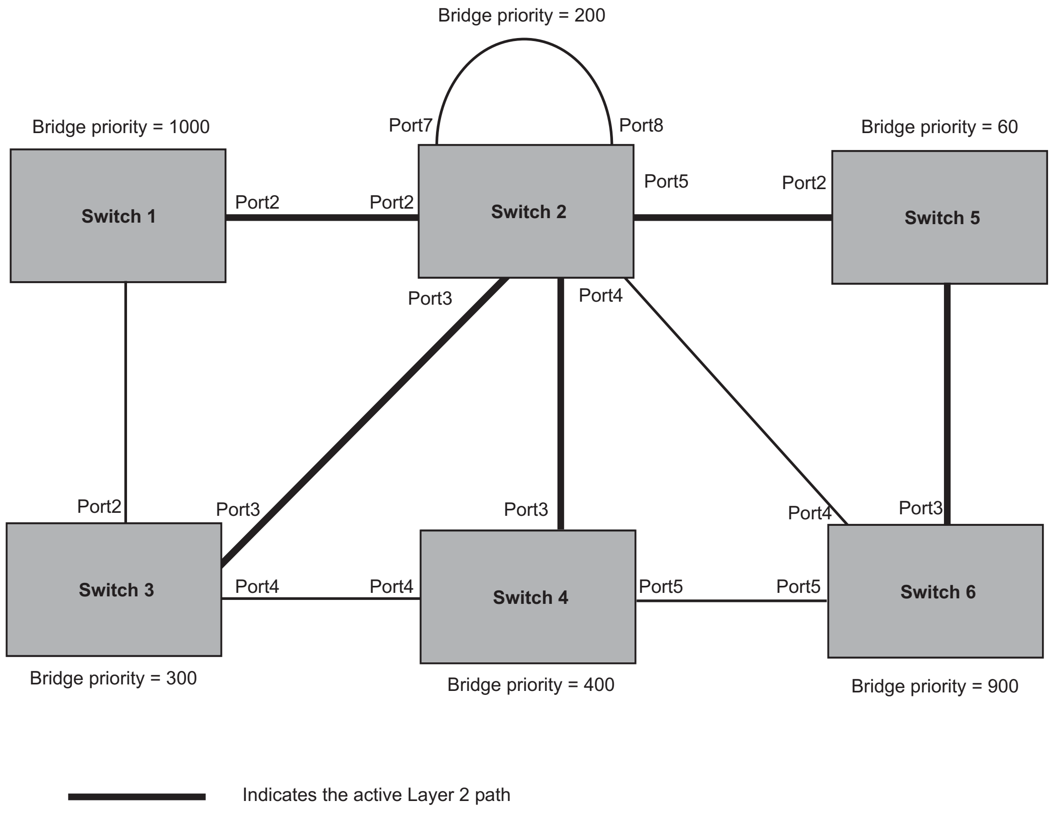
Port4/Switch 3 receives an RST BPDU that is inferior to what it can transmit; therefore, the port retains its Designated port role.

Ports on all the bridges in the topology with Designated port roles that received RST BPDUs with agreed flags go into forwarding states instantly. However, Designated ports that did not receive RST BPDUs with agreed flags must wait until the forward delay timer expires twice on those port. Only then will these port move into forwarding states.

The entire 802.1W topology converges in less than 300 msec and the essential connectivity is established between the designated ports and their connected root ports.

After convergence is complete, Figure 9.19 shows the active Layer 2 path of the topology in Figure 9.18.

Figure 9.19 Active Layer 2 Path in Complex Topology



Propagation of Topology Change

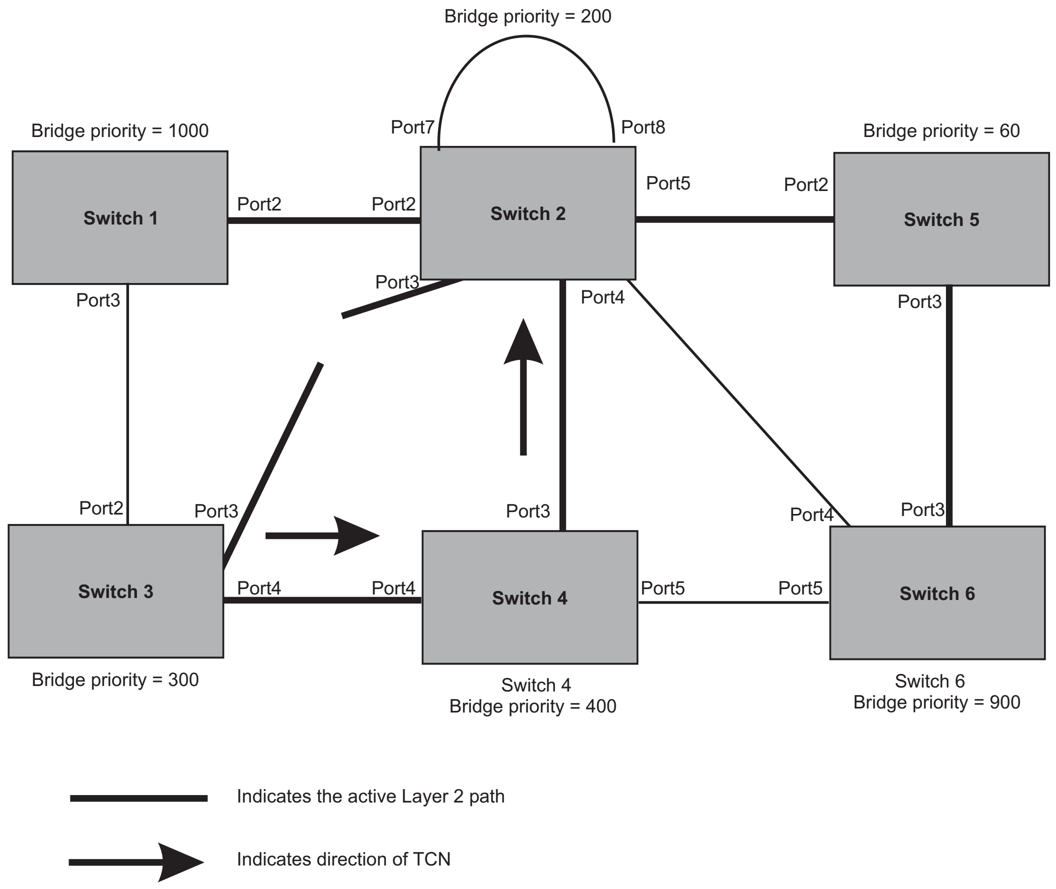
The Topology Change state machine generates and propagates the topology change notification messages on each port. When a Root port or a Designated port goes into a forwarding state, the Topology Change state machine on those ports send a topology change notice (TCN) to all the bridges in the topology to propagate the topology change.

NOTE: Edge ports, Alternate ports, or Backup ports do not need to propagate a topology change.

The TCN is sent in the RST BPDU that a port sends. Ports on other bridges in the topology then acknowledge the topology change once they receive the RST BPDU, and send the TCN to other bridges until all the bridges are informed of the topology change.

For example, Port3/Switch 2 in Figure 9.20, fails. Port4/Switch 3 becomes the new Root port. Port4/Switch 3 sends an RST BPDU with a TCN to Port4/Switch 4. To propagate the topology change, Port4/Switch 4 then starts a TCN timer on itself, on the bridge's Root port, and on other ports on that bridge with a Designated role. Then Port3/Switch 4 sends RST BPDU with the TCN to Port4/Switch 2. (Note the new active Layer 2 path in Figure 9.20.)

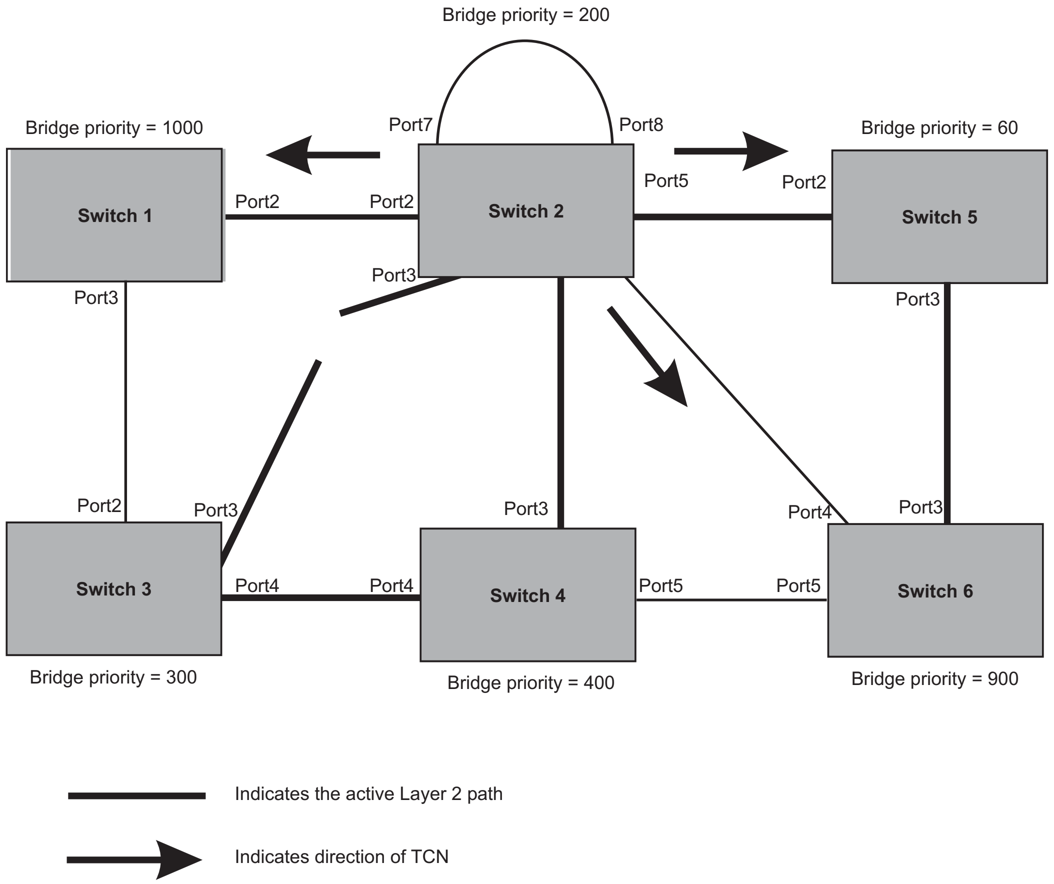
Figure 9.20 Beginning of Topology Change Notice



Switch 2 then starts the TCN timer on the Designated ports and sends RST BPDUs that contain the TCN as follows (Figure 9.21):

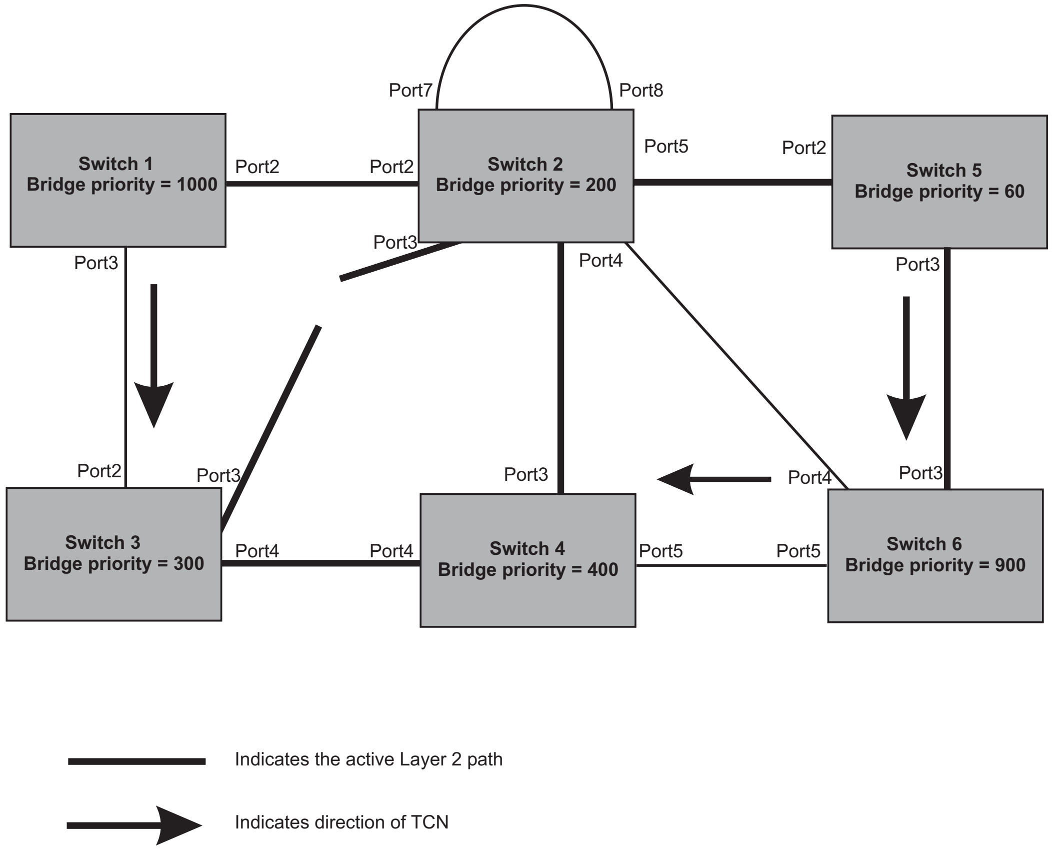
- Port5/Switch 2 sends the TCN to Port2/Switch 5
- Port4/Switch 2 sends the TCN to Port4/Switch 6
- Port2/Switch 2 sends the TCN to Port2/Switch 1

Figure 9.21 Sending TCN to Bridges Connected to Switch 2



Then Switch 1, Switch 5, and Switch 6 send RST BPDUs that contain the TCN to Switch 3 and Switch 4 to complete the TCN propagation (Figure 9.22).

Figure 9.22 Completing the TCN Propagation



Compatibility of 802.1W with 802.1D

802.1W-enabled bridges are backward compatible with IEEE 802.1D bridges. This compatibility is managed on a per-port basis by the Port Migration state machine. **However, intermixing the two types of bridges in the network topology is not advisable if you want to take advantage of the rapid convergence feature.**

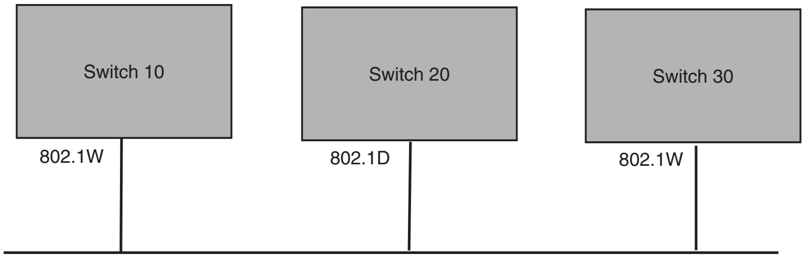
Compatibility with 802.1D means that an 802.1W-enabled port can send BPDUs in the STP or 802.1D format when one of the following events occur:

- The port receives a legacy BPDU. A legacy BPDU is an STP BPDU or a BPDU in an 802.1D format. The port that receives the legacy BPDU automatically configures itself to behave like a legacy port. It sends and receives legacy BPDUs only.
- The entire bridge is configured to operate in an 802.1D mode when an administrator sets the bridge parameter to zero at the CLI, forcing all ports on the bridge to send legacy BPDUs only.

Once a port operates in the 802.1D mode, 802.1D convergence times are used and rapid convergence is not realized.

For example, in Figure 9.23, Switch 10 and Switch 30 receive legacy BPDUs from Switch 20. Ports on Switch 10 and Switch 30 begin sending BPDUs in STP format to allow them to operate transparently with Switch 20.

Figure 9.23 802.1W Bridges with an 802.1D Bridge



Once Switch 20 is removed from the LAN, Switch 10 and Switch 30 receive and transmit BPDUs in the STP format to and from each other. This state will continue until the administrator enables the **force-migration-check** command to force the bridge to send RSTP BPDU during a migrate time period. If ports on the bridges continue to hear only STP BPDUs after this migrate time period, those ports will return to sending STP BPDUs. However, when the ports receive RST BPDUs during the migrate time period, the ports begin sending RST BPDUs. The migrate time period is non-configurable. It has a value of three seconds.

NOTE: The IEEE standards state that 802.1W bridges need to interoperate with 802.1D bridges. IEEE standards set the path cost of 802.1W bridges to be between 1 and 200,000,000; whereas path cost of 802.1D bridges are set between 1 and 65,535. In order for the two bridge types to be able to interoperate in the same topology, the administrator needs to configure the bridge path cost appropriately. Path costs for either 802.1W bridges or 802.1D bridges need to be changed; in most cases, path costs for 802.1W bridges need to be changed.

Configuring 802.1W Parameters on a Foundry Device

The remaining 802.1W sections explain how to configure the 802.1W protocol in a Foundry device.

NOTE: With RSTP running, enabling static trunk on ports that are members of VLAN 4000 will keep the system busy for 20 to 25 seconds.

Foundry devices are shipped from the factory with 802.1W disabled. Use the following methods to enable or disable 802.1W. You can enable or disable 802.1W at the following levels:

- Port-based VLAN – Affects all ports within the specified port-based VLAN. When you enable or disable 802.1W within a port-based VLAN, the setting overrides the global setting. Thus, you can enable 802.1W for the ports within a port-based VLAN even when 802.1W is globally disabled, or disable the ports within a port-based VLAN when 802.1W is globally enabled.
- Individual port – Affects only the individual port. However, if you change the 802.1W state of the primary port in a trunk group, the change affects all ports in the trunk group.

Enabling or Disabling 802.1W in a Port-Based VLAN

Use the following procedure to disable or enable 802.1W on a device on which you have configured a port-based VLAN. Changing the 802.1W state in a VLAN affects only that VLAN.

To enable 802.1W for all ports in a port-based VLAN, enter commands such as the following:

```
FastIron(config)#vlan 10
FastIron(config-vlan-10)#spanning-tree 802-1w
```

Syntax: [no] spanning-tree 802-1w

Note Regarding Pasting 802.1W Settings into the Running Configuration

If you paste 802.1W settings into the running configuration, and the pasted configuration includes ports that are already up, the ports will initially operate in STP legacy mode before operating in 802.1W RSTP mode. For example, the following pasted configuration will cause ports e 1 and e 2 to temporarily operate in STP legacy mode, because these ports are already up and running.

```
conf t
vlan 120
tag e 1 to e 2
spanning-tree 802-1w
spanning-tree 802-1w priority 1001
end
```

To avoid this issue, 802.1W commands/settings that are pasted into the configuration should be in the following order:

1. Ports that are not yet connected
2. 802.1W RSTP settings
3. Ports that are already up

For example:

```
conf t
vlan 120
untag e 3
spanning-tree 802-1w
spanning-tree 802-1w priority 1001
tag e 1 to 2
end
```

In the above configuration, untagged port e 3 is added to VLAN 120 *before* the 802.1W RSTP settings, and ports e 1 and e 2 are added *after* the 802.1W RSTP settings. When these commands are pasted into the running configuration, the ports will properly operate in 802.1W RSTP mode.

Enabling or Disabling 802.1W on a Single Spanning Tree

To enable 802.1W for all ports of a single spanning tree, enter a command such as the following:

```
FastIron(config-vlan-10)#spanning-tree single 802-1w
```

Syntax: [no] spanning-tree single 802-1w

Disabling or Enabling 802.1W on an Individual Port

The **spanning-tree 802-1w** or **spanning-tree single 802-1w** command must be used to initially enable 802.1W on ports. Both commands enable 802.1W on all ports that belong to the VLAN or to the single spanning tree.

Once 802.1W is enabled on a port, it can be disabled on individual ports. 802.1W that have been disabled on individual ports can then be enabled as required.

NOTE: If you change the 802.1W state of the primary port in a trunk group, the change affects all ports in that trunk group.

To disable or enable 802.1W on an individual port, enter commands such as the following:

```
FastIron(config)#interface e 1
FastIron(config-if-e1000-1)#no spanning-tree
```

Syntax: [no] spanning-tree

Changing 802.1W Bridge Parameters

When you make changes to 802.1W bridge parameters, the changes are applied to individual ports on the bridge. To change 802.1W bridge parameters, use the following methods.

To designate a priority for a bridge, enter a command such as the following:

```
FastIron(config)#spanning-tree 802-1w priority 10
```

The command in this example changes the priority on a device on which you have not configured port-based VLANs. The change applies to the default VLAN. If you have configured a port-based VLAN on the device, you can configure the parameters only at the configuration level for individual VLANs. Enter commands such as the following:

```
FastIron(config)#vlan 20
FastIron(config-vlan-20)#spanning-tree 802-1w priority 0
```

To make this change in the default VLAN, enter the following commands:

```
FastIron(config)#vlan 1
FastIron(config-vlan-1)#spanning-tree 802-1w priority 0
```

Syntax: spanning-tree 802-1w [forward-delay <value>] | [hello-time <value>] | [max-age <time>] | [force-version <value>] | [priority <value>]

The **forward-delay** <value> parameter specifies how long a port waits before it forwards an RST BPDU after a topology change. This can be a value from 4 – 30 seconds. The default is 15 seconds.

The **hello-time** <value> parameter specifies the interval between two hello packets. This parameter can have a value from 1 – 10 seconds. The default is 2 seconds.

The **max-age** <value> parameter specifies the amount of time the device waits to receive a hello packet before it initiates a topology change. You can specify a value from 6 – 40 seconds. The default is 20 seconds.

The value of **max-age** must be greater than the value of **forward-delay** to ensure that the downstream bridges do not age out faster than the upstream bridges (those bridges that are closer to the root bridge).

The **force-version** <value> parameter forces the bridge to send BPDUs in a specific format. You can specify one of the following values:

- 0 – The STP compatibility mode. Only STP (or legacy) BPDUs will be sent.
- 2 – The default. RST BPDUs will be sent unless a legacy bridge is detected. If a legacy bridge is detected, STP BPDUs will be sent instead.

The default is 2.

The **priority** <value> parameter specifies the priority of the bridge. You can enter a value from 0 – 65535. A lower numerical value means the bridge has a higher priority. Thus, the highest priority is 0. The default is 32768.

You can specify some or all of these parameters on the same command line. If you specify more than one parameter, you must specify them in the order shown above, from left to right.

Changing Port Parameters

The 802.1W port commands can be enabled on individual ports or on multiple ports, such as all ports that belong to a VLAN.

The 802.1W port parameters are preconfigured with default values. If the default parameters meet your network requirements, no other action is required.

You can change the following 802.1W port parameters using the following method.

```
FastIron(config)#vlan 10
FastIron(config-vlan-10)#spanning-tree 802-1w ethernet 5 path-cost 15 priority 64
```

Syntax: spanning-tree 802-1w ethernet [<slotnum>/]<portnum> path-cost <value> | priority <value> | [admin-edge-port] | [admin-pt2pt-mac] | [force-migration-check]

The <portnum> parameter specifies the interface used. If you are configuring a chassis device, specify the slot number as well as the port number (<slotnum>/<portnum>).

The **path-cost** <value> parameter specifies the cost of the port's path to the root bridge. 802.1W prefers the path with the lowest cost. You can specify a value from 1 – 20,000,000. Table 9.6 shows the recommended path cost values from the IEEE standards.

Table 9.6: Recommended Path Cost Values of 802.1W

Link Speed	Recommended (Default) 802.1W Path Cost Values	Recommended 802.1W Patch Cost Range
Less than 100 kilobits per second	200,000,000	20,000,000 – 200,000,000
1 Megabit per second	20,000,000	2,000,000 – 200,000,000
10 Megabits per second	2,000,000	200,000 – 200,000,000
100 Megabits per second	200,000	20,000 – 200,000,000
1 Gigabit per second	20,000	2,000 – 200,000,000
10 Gigabits per second	2,000	200 – 20,000
100 Gigabits per second	200	20 – 2,000
1 Terabits per second	20	2 – 200
10 Terabits per second	2	1 – 20

The **priority** <value> parameter specifies the preference that 802.1W gives to this port relative to other ports for forwarding traffic out of the topology. Y

- In releases prior to 03.0.00, you can specify a value from 8 – 252, in increments of 4. If you enter a value that is not divisible by four the software rounds to the nearest value that is. The default is 128. A higher numerical value means a lower priority; thus, the highest priority is 8.
- Starting in software release 03.0.00, you can specify a value from 0 – 240, in increments of 16. If you enter a value that is not divisible by 16, the software returns an error message. The default value is 128. A higher numerical value means a lower priority; thus, the highest priority is 0.

Set the **admin-edge-port** to enabled or disabled. If set to enabled, then the port becomes an edge port in the domain.

Set the **admin-pt2pt-mac** to enabled or disabled. If set to enabled, then a port is connected to another port through a point-to-point link. The point-to-point link increases the speed of convergence. This parameter, however, does not auto-detect whether or not the link is a physical point-to-point link.

The **force-migration-check** parameter forces the specified port to sent one RST BPDU. If only STP BPDUs are received in response to the sent RST BPDU, then the port will go return to sending STP BPDUs.

EXAMPLES:

Suppose you want to enable 802.1W on a system with no active port-based VLANs and change the hello-time from the default value of 2 to 8 seconds. Additionally, suppose you want to change the path and priority costs for port 5 only. To do so, enter the following commands.

```
FastIron(config)#spanning-tree 802-1w hello-time 8
FastIron(config)#spanning-tree 802-1w ethernet 5 path-cost 15 priority 64
```

Displaying Information about 802-1W

To display a summary of 802-1W, use the following command:

```
FastIron#show 802-1w
--- VLAN 1 [ STP Instance owned by VLAN 1 ] ----------------------------
VLAN 1 BPDU cam_index is 2 and the IGC and DMA master Are(HEX) 0 1 2 3
Bridge IEEE 802.1W Parameters:
Bridge            Bridge  Bridge Bridge Force   tx
Identifier        MaxAge  Hello  FwdDly Version Hold
hex               sec     sec    sec            cnt
800000e080541700  20      2      15     Default 3

RootBridge        RootPath   DesignatedBri-    Root  Max  Fwd  Hel
Identifier        Cost       dge Identifier    Port  Age  Dly  lo
hex                          hex                     sec  sec  sec
800000e0804c9c00  200000     800000e0804c9c00  1     20   15   2

Port IEEE 802.1W Parameters:
      <--- Config Params -->|<-------------- Current state ----------------->
Port  Pri PortPath P2P Edge Role       State      Designa- Designated
Num       Cost     Mac Port                       ted cost bridge
1     128 200000   F   F    ROOT       FORWARDING 0        800000e0804c9c00
2     128 200000   F   F    DESIGNATED FORWARDING 200000   800000e080541700
3     128 200000   F   F    DESIGNATED FORWARDING 200000   800000e080541700
4     128 200000   F   F    BACKUP     DISCARDING 200000   800000e080541700
```

Syntax: show 802-1w [vlan <vlan-id>]

The **vlan** <vlan-id> parameter displays 802.1W information for the specified port-based VLAN.

The **show 802.1w display** command shows the information listed in Table 9.7.

Table 9.7: CLI Display of 802.1W Summary

This Field...	Displays...
VLAN ID	The port-based VLAN that owns the STP instance. VLAN 1 is the default VLAN. If you have not configured port-based VLANs on this device, all 802.1W information is for VLAN 1.
Bridge IEEE 802.1W Parameters	
Bridge Identifier	The ID of the bridge.
Bridge Max Age	The configured max age for this bridge. The default is 20.
Bridge Hello	The configured hello time for this bridge.The default is 2.
Bridge FwdDly	The configured forward delay time for this bridge. The default is 15.
Force-Version	The configured force version value. One of the following value is displayed: • 0 – The bridge has been forced to operate in an STP compatibility mode. • 2 – The bridge has been forced to operate in an 802.1W mode. (This is the default.)

Table 9.7: CLI Display of 802.1W Summary (Continued)

This Field...	Displays...
txHoldCnt	The number of BPDUs that can be transmitted per Hello Interval. The default is 3.
Root Bridge Identifier	ID of the Root bridge that is associated with this bridge
Root Path Cost	The cost to reach the root bridge from this bridge. If the bridge is the root bridge, then this parameter shows a value of zero.
Designated Bridge Identifier	The bridge from where the root information was received.It can be from the root bridge itself, but it could also be from another bridge.
Root Port	The port on which the root information was received. This is the port that is connected to the Designated Bridge.
Max Age	The ***max age*** is derived from the Root port. An 802.1W-enabled bridge uses this value, along with the hello and message age parameters to compute the effective age of an RST BPDU. The ***message age*** parameter is generated by the Designated port and transmitted in the RST BPDU. RST BPDUs transmitted by a Designated port of the root bridge contains a message value of zero. ***Effective age*** is the amount of time the Root port, Alternate port, or Backup port retains the information it received from its peer Designated port. Effective age is reset every time a port receives an RST BPDU from its Designated port. If a Root port does not receive an RST BPDU from its peer Designated port for a duration more than the effective age, the Root port ages out the existing information and recomputes the topology. If the port is operating in 802.1D compatible mode, then max age functionality is the same as in 802.1D (STP).
Fwd Dly	The number of seconds a non-edge Designated port waits until it can apply any of the following transitions, if the RST BPDU it receives does not have an agreed flag: • Discarding state to learning state • Learning state to forwarding state When a non-edge port receives the RST BPDU it goes into forwarding state within 4 seconds or after two hello timers expire on the port. Fwd Dly is also the number of seconds that a Root port waits for an RST BPDU with a proposal flag before it applies the state transitions listed above. If the port is operating in 802.1D compatible mode, then forward delay functionality is the same as in 802.1D (STP).
Hello	The hello value derived from the Root port. It is the number of seconds between two Hello packets.
Port IEEE 802.1W Parameters	
Port Num	The port number shown in a slot#/port#format.
Pri	The configured priority of the port. The default is 128 or 0x80.
Port Path Cost	The configured path cost on a link connected to this port.

Table 9.7: CLI Display of 802.1W Summary (Continued)

This Field...	Displays...
P2P Mac	Indicates if the point-to-point-mac parameter is configured to be a point-to-point link: • T – The link is configured as a point-to-point link. • F – The link is not configured as a point-to-point link. This is the default.
Edge port	Indicates if the port is configured as an operational Edge port: • T – The port is configured as an Edge port. • F – The port is not configured as an Edge port. This is the default.
Role	The current role of the port: • Root • Designated • Alternate • Backup • Disabled Refer to “Bridges and Bridge Port Roles” on page 9-19 for definitions of the roles.
State	The port’s current 802.1W state. A port can have one of the following states: • Forwarding • Discarding • Learning • Disabled Refer to “Bridge Port States” on page 9-22 and “Edge Port and Non-Edge Port States” on page 9-22.
Designated Cost	The best root path cost that this port received, including the best root path cost that it can transmit.
Designated Bridge	The ID of the bridge that sent the best RST BPDU that was received on this port.

To display detailed information about 802-1W, using the following command:

```
FastIron#show 802-1w detail
======================================================================
VLAN 1 - MULTIPLE SPANNING TREE (MSTP - IEEE 802.1W) ACTIVE
======================================================================
BridgeId 800000e080541700, forceVersion 2, txHoldCount 3
Port 1 - Role: ROOT - State: FORWARDING
  PathCost 200000, Priority 128, AdminOperEdge F, AdminPt2PtMac F
  DesignatedPriority - Root: 0x800000e0804c9c00, Bridge: 0x800000e080541700
  ActiveTimers - rrWhile 4 rcvdInfoWhile 4
  MachineStates - PIM: CURRENT, PRT: ROOT_PORT, PST: FORWARDING
  TCM: ACTIVE, PPM: SENDING_STP, PTX: TRANSMIT_IDLE
  Received - RST BPDUs 0, Config BPDUs 1017, TCN BPDUs 0

Port 2 - Role: DESIGNATED - State: FORWARDING
  PathCost 200000, Priority 128, AdminOperEdge F, AdminPt2PtMac F
  DesignatedPriority - Root: 0x800000e0804c9c00, Bridge: 0x800000e080541700
  ActiveTimers - helloWhen 0
  MachineStates - PIM: CURRENT, PRT: DESIGNATED_PORT, PST: FORWARDING
  TCM: ACTIVE, PPM: SENDING_RSTP, PTX: TRANSMIT_IDLE
  Received - RST BPDUs 0, Config BPDUs 0, TCN BPDUs 0
```

Syntax: show 802-1w detail [vlan <vlan-id>]

The **vlan** <vlan-id> parameter displays 802.1W information for the specified port-based VLAN.

The **show spanning-tree 802.1W** command shows the following information.

Table 9.8: CLI Display of show spanning-tree 802.1W

This Field...	Displays...
VLAN ID	ID of the VLAN that owns the instance of 802.1W and whether or not it is active.
Bridge ID	ID of the bridge.
forceVersion	the configured version of the bridge: • 0 – The bridge has been forced to operate in an STP compatible mode. • 2 – The bridge has been forced to operate in an 802.1W mode.
txHoldCount	The number of BPDUs that can be transmitted per Hello Interval. The default is 3.
Port	ID of the port in slot#/port#format.

Table 9.8: CLI Display of show spanning-tree 802.1W (Continued)

This Field...	Displays...
Role	The current role of the port: • Root • Designated • Alternate • Backup • Disabled Refer to “Bridges and Bridge Port Roles” on page 9-19 for definitions of the roles.
State	The port’s current 802.1W state. A port can have one of the following states: • Forwarding • Discarding • Learning • Disabled Refer to “Bridge Port States” on page 9-22 and “Edge Port and Non-Edge Port States” on page 9-22.
Path Cost	The configured path cost on a link connected to this port.
Priority	The configured priority of the port. The default is 128 or 0x80.
AdminOperEdge	Indicates if the port is an operational Edge port. Edge ports may either be auto-detected or configured (forced) to be Edge ports using the CLI: • T – The port is and Edge port. • F – The port is not an Edge port. This is the default.
AdminP2PMac	Indicates if the point-to-point-mac parameter is configured to be a point-to-point link: • T – The link is a point-to-point link • F – The link is not a point-to-point link. This is the default.
DesignatedPriority	Shows the following: • Root – Shows the ID of the root bridge for this bridge. • Bridge – Shows the ID of the Designated bridge that is associated with this port.

Table 9.8: CLI Display of show spanning-tree 802.1W (Continued)

This Field...	Displays...
ActiveTimers	Shows what timers are currently active on this port and the number of seconds they have before they expire: • rrWhile – Recent root timer. A non-zero value means that the port has recently been a Root port. • rcvdInfoWhile – Received information timer. Shows the time remaining before the information held by this port expires (ages out). This timer is initialized with the effective age parameter. (See “Max Age” on page 9-49.) • rbWhile – Recent backup timer. A non-zero value means that the port has recently been a Backup port. • helloWhen – Hello period timer. The value shown is the amount of time between hello messages. • tcWhile – Topology change timer. The value shown is the interval when topology change notices can be propagated on this port. • fdWhile – Forward delay timer. (See the explanation Table on page 9-49.) • mdelayWhile – Migration delay timer. The amount of time that a bridge on the same LAN has to synchronize its migration state with this port before another BPDU type can cause this port to change the BPDU that it transmits.
Machine States	The current states of the various state machines on the port: • PIM – State of the Port Information state machine. • PRT – State of the Port Role Transition state machine. • PST – State of the Port State Transition state machine. • TCM – State of the Topology Change state machine. • PPM – State of the Port Protocol Migration. • PTX – State of the Port Transmit state machine. Refer to the section “State Machines” on page 9-23 for details on state machines.
Received	Shows the number of BPDU types the port has received: • RST BPDU – BPDU in 802.1W format. • Config BPDU – Legacy configuration BPDU (802.1D format). • TCN BPDU – Legacy topology change BPDU (802.1D format).

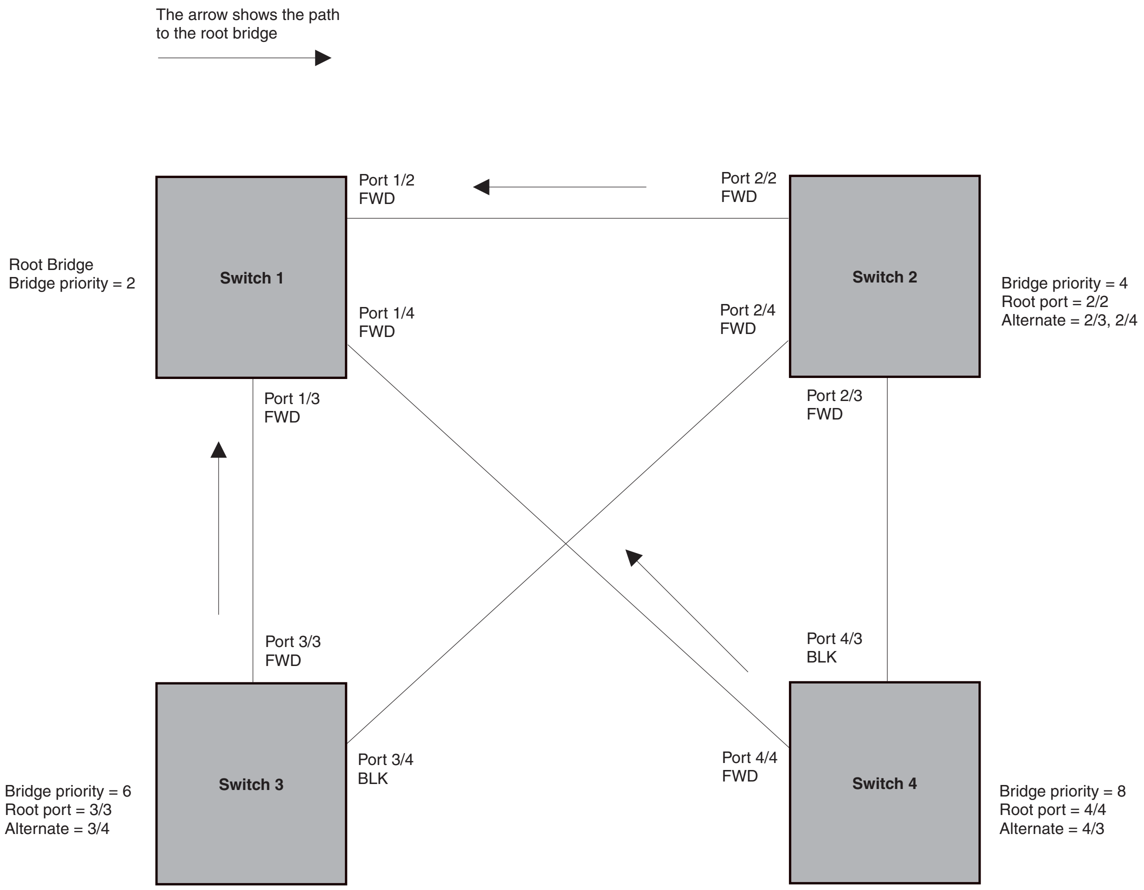
802.1W Draft 3

As an alternative to full 802.1W, you can configure 802.1W Draft 3. 802.1W Draft 3 provides a subset of the RSTP capabilities described in the 802.1W STP specification.

802.1W Draft 3 support is disabled by default. When the feature is enabled, if a root port on a Foundry device that is not the root bridge becomes unavailable, the device can automatically Switch over to an alternate root port, without reconvergence delays. 802.1W Draft 3 does not apply to the root bridge, since all the root bridge’s ports are always in the forwarding state.

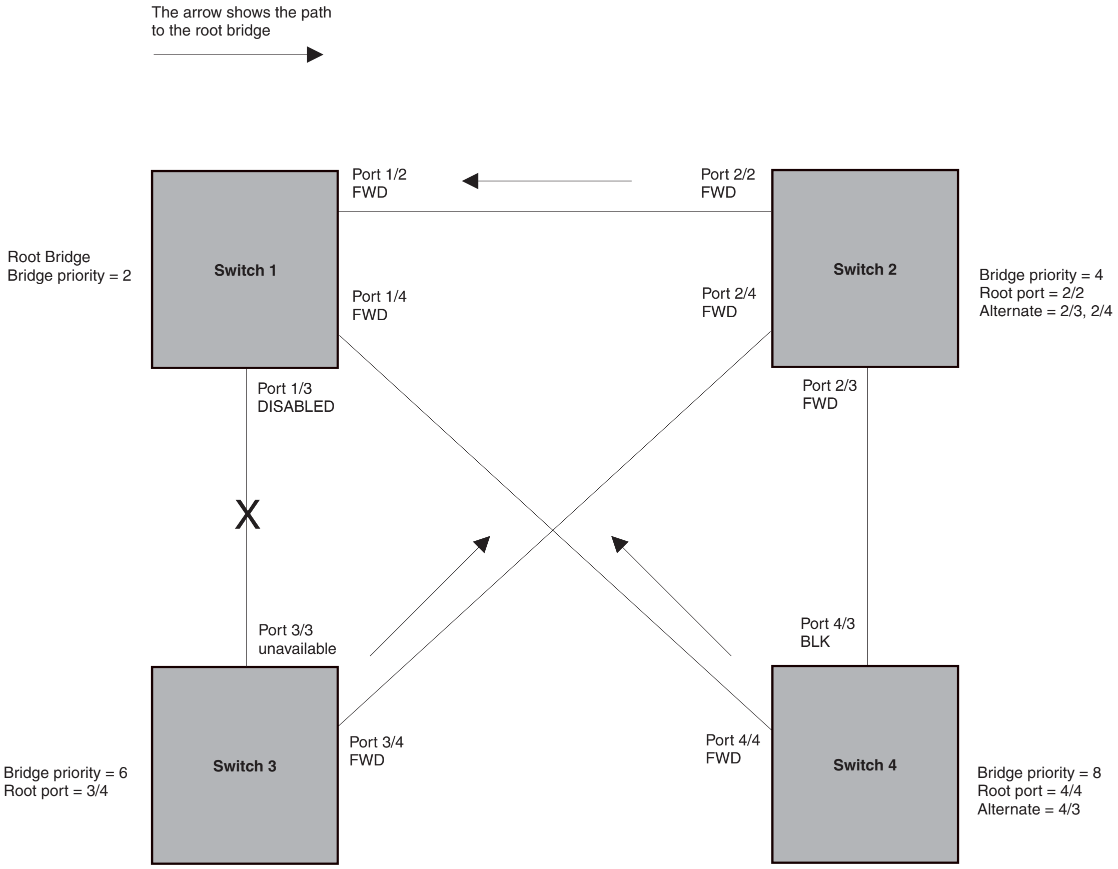
Figure 9.24 shows an example of an optimal STP topology. In this topology, all the non-root bridges have at least two paths to the root bridge (Switch 1 in this example). One of the paths is through the root port. The other path is a backup and is through the alternate port. While the root port is in the forwarding state, the alternate port is in the blocking state.

Figure 9.24 802.1W Draft 3 RSTP Ready for Failover



If the root port on a Switch becomes unavailable, 802.1W Draft 3 immediately fails over to the alternate port, as shown in Figure 9.25.

Figure 9.25 802.1W Draft 3 RSTP Failover to Alternate Root Port



In this example, port 3/3 on Switch 3 has become unavailable. In standard STP (802.1D), if the root port becomes unavailable, the Switch must go through the listening and learning stages on the alternate port to reconverge with the spanning tree. Thus, port 3/4 must go through the listening and learning states before entering the forwarding state and thus reconverging with the spanning tree.

802.1W Draft 3 avoids the reconvergence delay by calculating an alternate root port, and immediately failing over to the alternate port if the root port becomes unavailable. The alternate port is in the blocking state as long as the root port is in the forwarding state, but moves immediately to the active state if the root port becomes unavailable. Thus, using 802.1W Draft 3, Switch 3 immediately fails over to port 3/4, without the delays caused by the listening and learning states.

802.1W Draft 3 selects the port with the next-best cost to the root bridge. For example, on Switch 3, port 3/3 has the best cost to the root bridge and thus is selected by STP as the root port. Port 3/4 has the next-best cost to the root bridge, and thus is selected by 802.1W Draft 3 as the alternate path to the root bridge.

Once a failover occurs, the Switch no longer has an alternate root port. If the port that was an alternate port but became the root port fails, standard STP is used to reconverge with the network. You can minimize the reconvergence delay in this case by setting the forwarding delay on the root bridge to a lower value. For example, if the forwarding delay is set to 15 seconds (the default), change the forwarding delay to a value from 3 – 10 seconds.

During failover, 802.1W Draft 3 flushes the MAC addresses leaned on the unavailable root port, selects the alternate port as the new root port, and places that port in the forwarding state. If traffic is flowing in both directions on the new root port, addresses are flushed (moved) in the rest of the spanning tree automatically.

Reconvergence Time

Spanning tree reconvergence using 802.1W Draft 3 can occur within one second.

After the spanning tree reconverges following the topology change, traffic also must reconverge on all the bridges attached to the spanning tree. This is true regardless of whether 802.1W Draft 3 or standard STP is used to reconverge the spanning tree.

Traffic reconvergence happens after the spanning tree reconvergence, and is achieved by flushing the Layer 2 information on the bridges.

- Following 802.1W Draft 3 reconvergence of the spanning tree, traffic reconvergence occurs in the time it takes for the bridge to detect the link changes plus the STP maximum age set on the bridge.
- If standard STP reconvergence occurs instead, traffic reconvergence takes two times the forward delay plus the maximum age.

NOTE: 802.1W Draft 3 does not apply when a failed root port comes back up. When this happens, standard STP is used.

Configuration Considerations

802.1W Draft 3 is disabled by default. To ensure optimal performance of the feature before you enable it:

- Configure the bridge priorities so that the root bridge is one that supports 802.1W Draft 3. (Use a Foundry device or third-party device that supports 802.1W Draft 3.)
- Change the forwarding delay on the root bridge to a value lower than the default 15 seconds. Foundry recommends a value from 3 – 10 seconds. The lower forwarding delay helps reduce reconvergence delays in cases where 802.1W Draft 3 is not applicable, such as when a failed root port comes back up.
- Configure the bridge priorities and root port costs so that each device has an active path to the root bridge if its root port becomes unavailable. For example, port 3/4 is connected to port 2/4 on Switch 2, which has the second most favorable bridge priority in the spanning tree.

NOTE: If reconvergence involves changing the state of a root port on a bridge that supports 802.1D STP but not 802.1W Draft 3, then reconvergence still requires the amount of time it takes for the ports on the 802.1D bridge to change state to forwarding (as needed), and receive BPDUs from the root bridge for the new topology.

Enabling 802.1W Draft 3

802.1W Draft 3 is disabled by default. The procedure for enabling the feature differs depending on whether single STP is enabled on the device.

NOTE: STP must be enabled before you can enable 802.1W Draft 3.

Enabling 802.1W Draft 3 When Single STP Is Not Enabled

By default, each port-based VLAN on the device has its own spanning tree. To enable 802.1W Draft 3 in a port-based VLAN, enter commands such as the following:

```
FastIron(config)#vlan 10
FastIron(config-vlan-10)#spanning-tree rstp
```

Syntax: [no] spanning-tree rstp

This command enables 802.1W Draft 3. You must enter the command separately in each port-based VLAN in which you want to run 802.1W Draft 3.

NOTE: This command does not also enable STP. To enable STP, first enter the **spanning-tree** command without the **rstp** parameter. After you enable STP, enter the **spanning-tree rstp** command to enable 802.1W Draft 3.

To disable 802.1W Draft 3, enter the following command:

```
FastIron(config-vlan-10)#no spanning-tree rstp
```

Enabling 802.1W Draft 3 When Single STP Is Enabled

To enable 802.1W Draft 3 on a device that is running single STP, enter the following command at the global CONFIG level of the CLI:

```
FastIron(config)#spanning-tree single rstp
```

Syntax: [no] spanning-tree single rstp

This command enables 802.1W Draft 3 on the whole device.

NOTE: This command does not also enable single STP. To enable single STP, first enter the **spanning-tree single** command without the **rstp** parameter. After you enable single STP, enter the **spanning-tree single rstp** command to enable 802.1W Draft 3.

To disable 802.1W Draft 3 on a device that is running single STP, enter the following command:

```
FastIron(config)#no spanning-tree single rstp
```

Single Spanning Tree (SSTP)

By default, each port-based VLAN on a Foundry device runs a separate spanning tree, which you can enable or disable on an individual VLAN basis.

Alternatively, you can configure a Foundry device to run a single spanning tree across all ports and VLANs on the device. The Single STP feature (SSTP) is especially useful for connecting a Foundry device to third-party devices that run a single spanning tree in accordance with the 802.1Q specification.

SSTP uses the same parameters, with the same value ranges and defaults, as the default STP support on Foundry devices. See “STP Parameters and Defaults” on page 9-2.

SSTP Defaults

SSTP is disabled by default. When you enable the feature, all VLANs on which STP is enabled become members of a single spanning tree. All VLANs on which STP is disabled are excluded from the single spanning tree.

- To add a VLAN to the single spanning tree, enable STP on that VLAN.
- To remove a VLAN from the single spanning tree, disable STP on that VLAN.

When you enable SSTP, all the ports that are in port-based VLANs with STP enabled become members of a single spanning tree domain. Thus, the ports share a single BPDU broadcast domain. The Foundry device places all the ports in a non-configurable VLAN, 4094, to implement the SSTP domain. However, this VLAN does not affect port membership in the port-based VLANs you have configured. Other broadcast traffic is still contained within the individual port-based VLANs. Therefore, you can use SSTP while still using your existing VLAN configurations without changing your network. In addition, SSTP does not affect 802.1Q tagging. Tagged and untagged ports alike can be members of the single spanning tree domain.

NOTE: When SSTP is enabled, the BPDUs on tagged ports go out untagged.

If you disable SSTP, all VLANs that were members of the single spanning tree run MSTP instead. In MSTP, each VLAN has its own spanning tree. VLANs that were not members of the single spanning tree were not enabled for STP. Therefore, STP remains disabled on those VLANs.

Enabling SSTP

To enable SSTP, use one of the following methods.

NOTE: If the device has only one port-based VLAN (the default VLAN), then the device is already running a single instance of STP. In this case, you do not need to enable SSTP. You need to enable SSTP only if the device contains more than one port-based VLAN and you want all the ports to be in the same STP broadcast domain.

To configure the Foundry device to run a single spanning tree, enter the following command at the global CONFIG level.

```
FastIron(config)#spanning-tree single
```

NOTE: If the device has only one port-based VLAN, the CLI command for enabling SSTP is not listed in the CLI. The command is listed only if you have configured a port-based VLAN.

To change a global STP parameter, enter a command such as the following at the global CONFIG level:

```
FastIron(config) spanning-tree single priority 2
```

This command changes the STP priority for all ports to 2.

To change an STP parameter for a specific port, enter commands such as the following:

```
FastIron(config) spanning-tree single ethernet 1 priority 10
```

The commands shown above override the global setting for STP priority and set the priority to 10 for port 1/1.

Here is the syntax for the global STP parameters.

Syntax: [no] spanning-tree single [forward-delay <value>]
[hello-time <value>] | [maximum-age <time>] | [priority <value>]

Here is the syntax for the STP port parameters.

Syntax: [no] spanning-tree single [ethernet [<slotnum>/]<portnum> path-cost <value> | priority <value>]

NOTE: Both commands listed above are entered at the global CONFIG level.

Displaying SSTP information

To verify that SSTP is in effect, enter the following commands at any level of the CLI:

```
FastIron#show span
```

Syntax: show span [vlan <vlan-id>] | [pvst-mode] | [<num>] |
[detail [vlan <vlan-id> [ethernet [<slotnum>/]<portnum>] | <num>]]

The **vlan** <vlan-id> parameter displays STP information for the specified port-based VLAN.

The **pvst-mode** parameter displays STP information for the device's Per VLAN Spanning Tree (PVST+) compatibility configuration. See "PVST/PVST+ Compatibility" on page 9-62.

The <num> parameter displays only the entries after the number you specify. For example, on a device with three port-based VLANs, if you enter 1, then information for the second and third VLANs is displayed, but information for the first VLAN is not displayed. Information is displayed according to VLAN number, in ascending order. The entry number is not the same as the VLAN number. For example, if you have port-based VLANs 1, 10, and 2024, then the command output has three STP entries. To display information for VLANs 10 and 2024 only, enter **show span 1**.

The **detail** parameter and its additional optional parameters display detailed information for individual ports. See "Displaying Detailed STP Information for Each Interface" on page 9-12.

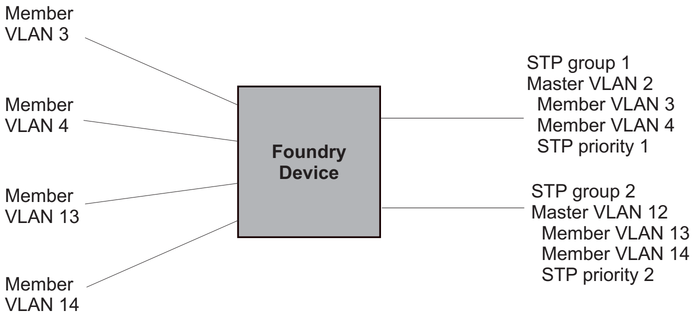
STP per VLAN Group

STP per VLAN group is an STP enhancement that provides scalability while overcoming the limitations of the following scalability alternatives:

- Standard STP – You can configure up to 254 instances of standard STP on a Foundry FastIron X Series device. It is possible to need more instances of STP than this in large configurations. Using STP per VLAN group, you can aggregate STP instances.
- Single STP – Single STP allows all the VLANs to run STP, but each VLAN runs the same instance of STP, resulting in numerous blocked ports that do not pass any Layer 2 traffic. STP per VLAN group uses all available links by load balancing traffic for different instances of STP on different ports. A port that blocks traffic for one spanning tree forwards traffic for another spanning tree.

STP per VLAN group allows you to group VLANs and apply the same STP parameter settings to all the VLANs in the group. Figure 9.26 shows an example of a STP per VLAN group implementation.

Figure 9.26 STP per VLAN Group Example



A master VLAN contains one or more member VLANs. Each of the member VLANs in the STP Group runs the same instance of STP and uses the STP parameters configured for the master VLAN. In this example, the Foundry device is configured with VLANs 3, 4, 13, and 14. VLANs 3 and 4 are grouped in master VLAN 2, which is in STP group 1. VLANs 13 and 14 are grouped in master VLAN 12, which is in STP group 2. The VLANs in STP group 1 all share the same spanning tree. The VLANs in STP group 2 share a different spanning tree.

All the ports in the VLANs are tagged. The ports must be tagged so that they can be in both a member VLAN and the member's master VLAN. For example, ports 1/1 – 1/4 are in member VLAN 3 and also in master VLAN 2 (since master VLAN 2 contains member VLAN 3).

STP Load Balancing

Notice that the STP groups each have different STP priorities. In configurations that use the STP groups on multiple devices, you can use the STP priorities to load balance the STP traffic. By setting the STP priorities for the same STP group to different values on each device, you can cause each of the devices to be the root bridge for a different STP group. This type of configuration distributes the traffic evenly across the devices and also ensures that ports that are blocked in one STP group's spanning tree are used by another STP group's spanning tree for forwarding. See “Configuration Example for STP Load Sharing” on page 9-61 for an example using STP load sharing.

Configuring STP per VLAN Group

To configure STP per VLAN group:

- Configure the member VLANs.
- Optionally, configure master VLANs to contain the member VLANs. This is useful when you have a lot of member VLANs and you do not want to individually configure STP on each one. Each of the member VLANs in the STP group uses the STP settings of the master VLAN.
- Configure the STP groups. Each STP group runs a separate instance of STP.

Here are the CLI commands for implementing the STP per VLAN group configuration shown in Figure 9.26. The following commands configure the member VLANs (3, 4, 13, and 14) and the master VLANs (2 and 12). Notice that changes to STP parameters are made in the master VLANs only, not in the member VLANs.

```
FastIron(config)#vlan 2
FastIron(config-vlan-2)#spanning-tree priority 1
FastIron(config-vlan-2)#tagged ethernet 1/1 to 1/4
FastIron(config-vlan-2)#vlan 3
FastIron(config-vlan-3)#tagged ethernet 1/1 to 1/4
FastIron(config-vlan-3)#vlan 4
FastIron(config-vlan-4)#tagged ethernet 1/1 to 1/4
FastIron(config-vlan-4)#vlan 12
FastIron(config-vlan-12)#spanning-tree priority 2
FastIron(config-vlan-12)#tagged ethernet 1/1 to 1/4
FastIron(config-vlan-12)#vlan 13
FastIron(config-vlan-13)#tagged ethernet 1/1 to 1/4
FastIron(config-vlan-13)#vlan 14
FastIron(config-vlan-14)#tagged ethernet 1/1 to 1/4
FastIron(config-vlan-14)#exit
```

The following commands configure the STP groups.

```
FastIron(config)#stp-group 1
FastIron(config-stp-group-1)#master-vlan 2
FastIron(config-stp-group-1)#member-vlan 3 to 4
FastIron(config-stp-group-1)#exit
FastIron(config)#stp-group 2
FastIron(config-stp-group-2)#master-vlan 12
FastIron(config-stp-group-2)#member-vlan 13 to 14
```

Syntax: [no] stp-group <num>

This command changes the CLI to the STP group configuration level. The following commands are valid at this level. The <num> parameter specifies the STP group ID and can be from 1 – 32.

Syntax: [no] master-vlan <num>

This command adds a master VLAN to the STP group. The master VLAN contains the STP settings for all the VLANs in the STP per VLAN group. The <num> parameter specifies the VLAN ID. An STP group can contain one master VLAN.

NOTE: If you delete the master VLAN from an STP group, the software automatically assigns the first member VLAN in the group to be the new master VLAN for the group.

Syntax: [no] member-vlan <num> [to <num>]

This command adds additional VLANs to the STP group. These VLANs also inherit the STP settings of the master VLAN in the group.

Syntax: [no] member-group <num>

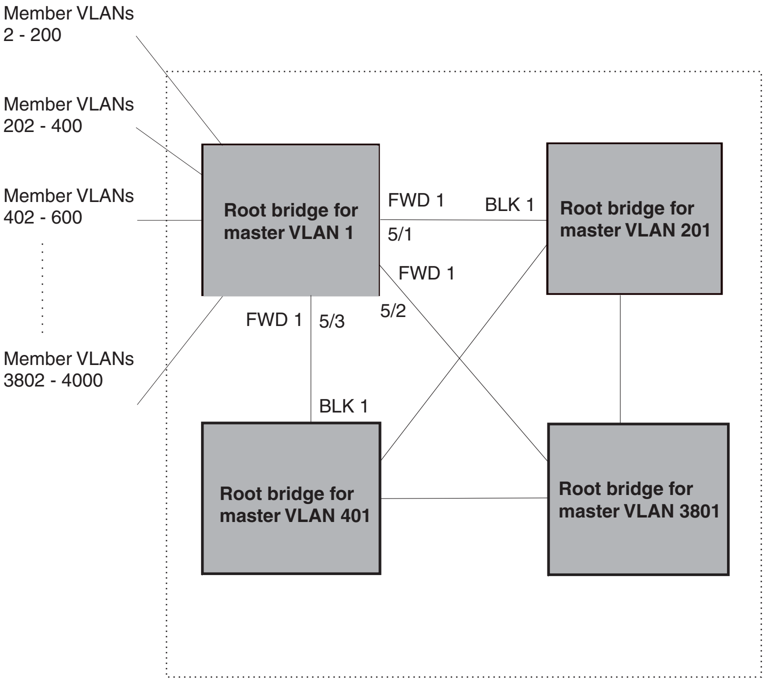
This command adds a member group (a VLAN group) to the STP group. All the VLANs in the member group inherit the STP settings of the master VLAN in the group. The <num> parameter specifies the VLAN group ID.

NOTE: This command is optional and is not used in the example above. For an example of this command, see “Configuration Example for STP Load Sharing” .

Configuration Example for STP Load Sharing

Figure 9.27 shows another example of a STP per VLAN group implementation.

Figure 9.27 More Complex STP per VLAN Group Example


Member VLANs
2 - 200
Member VLANs
202 - 400
Member VLANs
402 - 600
Member VLANs
3802 - 4000
Root bridge for master VLAN 1
FWD 1
BLK 1
5/1
FWD 1
5/2
FWD 1
5/3
BLK 1
Root bridge for master VLAN 201
Root bridge for master VLAN 401
Root bridge for master VLAN 3801

In this example, each of the devices in the core is configured with a common set of master VLANs, each of which contains one or more member VLANs. Each of the member VLANs in an STP group runs the same instance of STP and uses the STP parameters configured for the master VLAN.

The STP group ID identifies the STP instance. All VLANs within an STP group run the same instance of STP. The master VLAN specifies the bridge STP parameters for the STP group, including the bridge priority. In this example, each of the devices in the core is configured to be the default root bridge for a different master VLAN. This configuration ensures that each link can be used for forwarding some traffic. For example, all the ports on the root bridge for master VLAN 1 are configured to forward BPDUs for master VLAN's spanning tree. Ports on the other devices block or forward VLAN 1's traffic based on STP convergence. All the ports on the root bridge for VLAN 2 forward VLAN 2's traffic, and so on.

All the ports in the VLANs are tagged. The ports must be tagged so that they can be in both a member VLAN and the member's master VLAN. For example, port 1/1 – and ports 5/1, 5/2, and 5/3 are in member VLAN 2 and master VLAN 1 (since master VLAN a contains member VLAN 2).

Here are the commands for configuring the root bridge for master VLAN 1 in figure Figure 9.26 for STP per VLAN group. The first group of commands configures the master VLANs. Notice that the STP priority is set to a different value for each VLAN. In addition, the same VLAN has a different STP priority on each device. This provides load balancing by making each of the devices a root bridge for a different spanning tree.

```
FastIron(config)#vlan 1
FastIron(config-vlan-1)#spanning-tree priority 1
FastIron(config-vlan-1)#tag ethernet 1/1 ethernet 5/1 to 5/3
FastIron(config-vlan-1)#vlan 201
FastIron(config-vlan-201)#spanning-tree priority 2
FastIron(config-vlan-201)#tag ethernet 1/2 ethernet 5/1 to 5/3
FastIron(config-vlan-201)#vlan 401
FastIron(config-vlan-401)#spanning-tree priority 3
FastIron(config-vlan-401)#tag ethernet 1/3 ethernet 5/1 to 5/3
...
```

```
FastIron(config-vlan-3601)#vlan 3801
FastIron(config-vlan-3801)#spanning-tree priority 20
FastIron(config-vlan-3801)#tag ethernet 1/20 ethernet 5/1 to 5/3
FastIron(config-vlan-3801)#exit
```

The next group of commands configures VLAN groups for the member VLANs. Notice that the VLAN groups do not contain the VLAN numbers assigned to the master VLANs. Also notice that no STP parameters are configured for the groups of member VLANs. Each group of member VLANs will inherit its STP settings from its master VLAN.

Set the bridge priority for each master VLAN to the highest priority (1) on one of the devices in the STP per VLAN group configuration. By setting the bridge priority to the highest priority, you make the device the default root bridge for the spanning tree. To ensure STP load balancing, make each of the devices the default root bridge for a different master VLAN.

```
FastIron(config)#vlan-group 1 vlan 2 to 200
FastIron(config-vlan-group-1)#tag ethernet 1/1 ethernet 5/1 to 5/3
FastIron(config-vlan-group-1)#vlan-group 2 vlan 202 to 400
FastIron(config-vlan-group-2)#tag ethernet 1/2 ethernet 5/1 to 5/3
FastIron(config-vlan-group-2)#vlan-group 3 vlan 402 to 600
FastIron(config-vlan-group-2)#tag ethernet 1/3 ethernet 5/1 to 5/3
...
FastIron(config-vlan-group-19)#vlan-group 20 vlan 3082 to 4000
FastIron(config-vlan-group-20)#tag ethernet 1/20 ethernet 5/1 to 5/3
FastIron(config-vlan-group-20)#exit
```

The following group of commands configures the STP groups. Each STP group in this configuration contains one master VLAN, which contains a VLAN group. This example shows that an STP group also can contain additional VLANs (VLANs not configured in a VLAN group).

```
FastIron(config)#stp-group 1
FastIron(config-stp-group-1)#master-vlan 1
FastIron(config-stp-group-1)#member-group 1
FastIron(config-stp-group-1)#member-vlan 4001 4004 to 4010
FastIron(config-stp-group-1)#stp-group 2
FastIron(config-stp-group-2)#master-vlan 201
FastIron(config-stp-group-2)#member-group 2
FastIron(config-stp-group-2)#member-vlan 4002 4003 4011 to 4015
FastIron(config-stp-group-2)#stp-group 3
FastIron(config-stp-group-3)#master-vlan 401
FastIron(config-stp-group-3)#member-group 3
...
FastIron(config-stp-group-19)#stp-group 20
FastIron(config-stp-group-20)#master-vlan 3081
FastIron(config-stp-group-20)#member-group 20
```

PVST/PVST+ Compatibility

The FastIron family of switches support Cisco's Per VLAN Spanning Tree plus (PVST+), by allowing the device to run multiple spanning trees (MSTP) while also interoperating with IEEE 802.1Q devices[1].

NOTE: Foundry ports automatically detect PVST+ BPDUs and enable support for the BPDUs once detected. You do not need to perform any configuration steps to enable PVST+ support. However, to support the IEEE 802.1Q BPDUs, you might need to enable dual-mode support.

---

[1] Cisco user documentation for PVST/PVST+ refers to the IEEE 802.1Q spanning tree as the ***Common Spanning Tree (CST)***.

Foundry's support for Cisco's Per VLAN Spanning Tree plus (PVST+), allows a Foundry device to run multiple spanning trees (MSTP) while also interoperating with IEEE 802.1Q devices. Foundry ports automatically detect PVST+ BPDUs and enable support for the BPDUs once detected. The enhancement allows a port that is in PVST+ compatibility mode due to auto-detection to revert to the default MSTP mode when one of the following events occurs:

- The link is disconnected or broken
- The link is administratively disabled
- The link is disabled by interaction with the link-keepalive protocol

This enhancement allows a port that was originally interoperating with PVST+ to revert to MSTP when connected to a Foundry device.

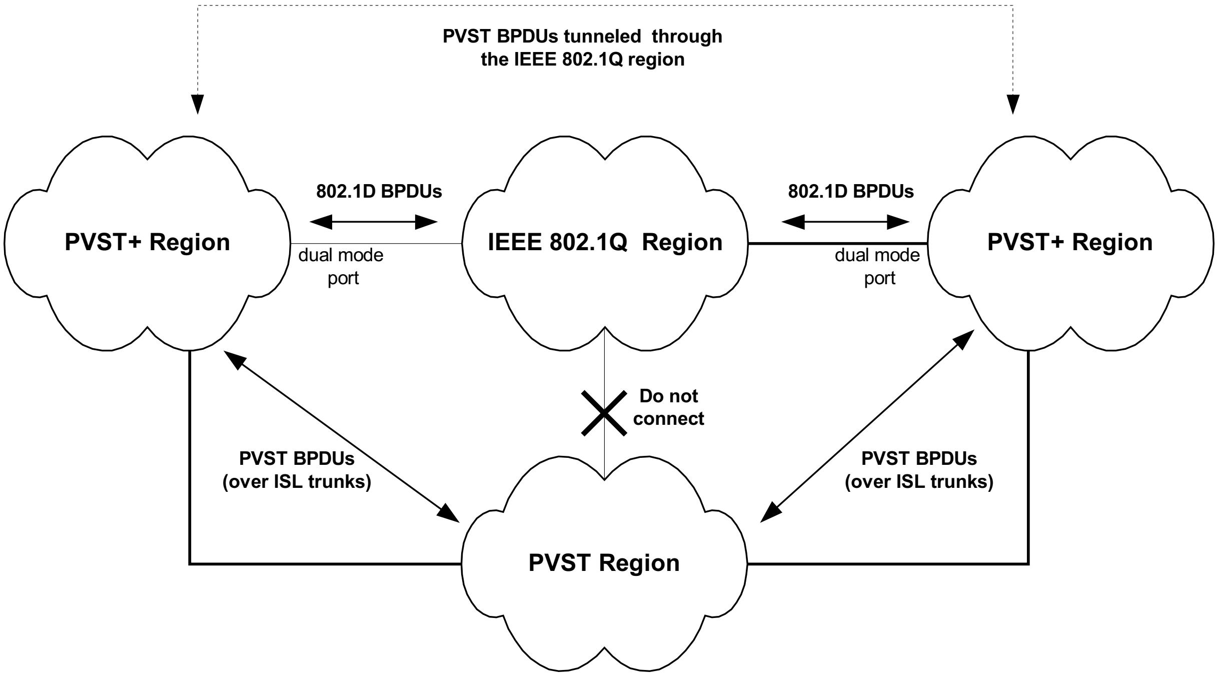
Overview of PVST and PVST+

Per VLAN Spanning Tree (PVST) is a Cisco proprietary protocol that allows a Cisco device to have multiple spanning trees. The Cisco device can interoperate with spanning trees on other PVST devices but cannot interoperate with IEEE 802.1Q devices. An IEEE 802.1Q device has all its ports running a single spanning tree. ***PVST+*** is an extension of PVST that allows a Cisco device to also interoperate with devices that are running a single spanning tree (IEEE 802.1Q).

The enhanced PVST+ support allows a Foundry device to interoperate with PVST spanning trees and the IEEE 802.1Q spanning tree at the same time.

IEEE 802.1Q and PVST regions cannot interoperate directly but can interoperate indirectly through PVST+ regions. PVST BPDUs are tunnelled through 802.1Q regions, while PVST BPDUs for VLAN 1 (the IEEE 802.1Q VLAN) are processed by PVST+ regions. Figure 9.28 shows the interaction of IEEE 802.1Q, PVST, and PVST+ regions.

Figure 9.28 Interaction of IEEE 802.1Q, PVST, and PVST+ Regions



VLAN Tags and Dual Mode

The ***dual-mode*** feature enables a port to send and receive both tagged and untagged frames. When the dual-mode feature is enabled on a port, the port is an untagged member of one of its VLANs and is at the same time a tagged member of all its other VLANs. The untagged frames are supported on the port's ***Port Native VLAN***.

The dual-mode feature must be enabled on a Foundry port in order to interoperate with another vendor's device. Some vendors use VLAN 1 by default to support the IEEE 802.1Q-based standard spanning tree protocols, such as 802.1d and 802.1w for sending untagged frames on VLAN 1. On Foundry FastIron switches, by default, the ***Port Native VLAN*** is the same as the ***Default VLAN***, which is VLAN 1. Thus, to support IEEE 802.1Q in a typical configuration, a port must be able to send and receive untagged frames for VLAN 1 and tagged frames for the other VLANs, and interoperate with other vendor's devices using VLAN 1.

If you want to use tagged frames on VLAN 1, you can change the default VLAN ID to an ID other than 1. You also can specify the VLAN on which you want the port to send and receive untagged frames (the Port Native VLAN). The Port Native VLAN ID does not need to be the same as the default VLAN. Make sure that the untagged (native) VLAN is also changed on the interoperating vendor side to match that on the Foundry side.

To support the IEEE 802.1Q with non-standard proprietary protocols such as PVST and PVST+, a port must always send and receive untagged frames on VLAN 1 on both sides. In this case, enable the dual-mode 1 feature to allow untagged BPDUs on VLAN 1and use Native VLAN 1 on the interoperating vendor side. You should not use VLAN 1 for tagged frames in this case.

Configuring PVST+ Support

PVST+ support is automatically enabled when the port receives a PVST BPDU. You can manually enable the support at any time or disable the support if desired.

If you want a tagged port to also support IEEE 802.1Q BPDUs, you need to enable the dual-mode feature on the port. The dual-mode feature is disabled by default and must be enabled manually.

A port that is in PVST+ compatibility mode due to auto-detection reverts to the default MSTP mode when one of the following events occurs:

- The link is disconnected or broken
- The link is administratively disabled
- The link is disabled by interaction with the link-keepalive protocol

This allows a port that was originally interoperating with PVST+ to revert to MSTP when connected to a Foundry device.

Enabling PVST+ Support Manually

To immediately enable PVST+ support on a port, enter commands such as the following:

```
FastIron(config)#interface ethernet 1/1
FastIron(config-if-1/1)#pvst-mode
```

Syntax: [no] pvst-mode

NOTE: If you disable PVST+ support, the software still automatically enables PVST+ support if the port receives a BPDU with PVST+ format.

NOTE: If 802.1W and pvst-mode (either by auto-detection or by explicit configuration) are enabled on a tagged VLAN port, 802.1W will treat the PVST BPDUs as legacy 802.1D BPDUs.

Enabling Dual-Mode Support

To enable the dual-mode feature on a port, enter the following command at the interface configuration level for the port:

```
FastIron(config-if-1/1)#dual-mode
```

Syntax: [no] dual-mode [<vlan-id>]

The <vlan-id> specifies the port's Port Native VLAN. This is the VLAN on which the port will support untagged frames. By default, the Port Native VLAN is the same as the default VLAN (which is VLAN 1 by default).

For more information about the dual-mode feature, see "Dual-Mode VLAN Ports" on page 14-57.

Displaying PVST+ Support Information

To display PVST+ information for ports on a Foundry device, enter the following command at any level of the CLI:

```
FastIron#show span pvst-mode
PVST+ Enabled on:
Port      Method
1/1       Set by configuration
1/2       Set by configuration
2/10      Set by auto-detect
3/12      Set by configuration
4/24      Set by auto-detect
```

Syntax: show span pvst-mode

This command displays the following information.

Table 9.9: CLI Display of PVST+ Information

This Field...	Displays...
Port	The Foundry port number. **Note**: The command lists information only for the ports on which PVST+ support is enabled.
Method	The method by which PVST+ support was enabled on the port. The method can be one of the following: • Set by configuration – You enabled the support. • Set by auto-detect – The support was enabled automatically when the port received a PVST+ BPDU.

Configuration Examples

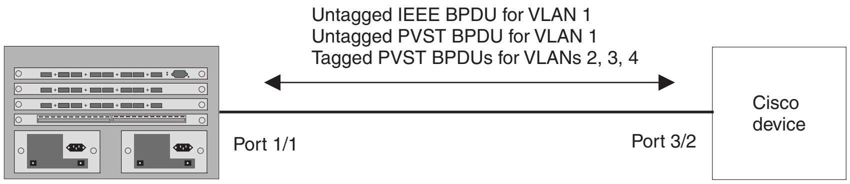
The following examples show configuration examples for two common configurations:

- Untagged IEEE 802.1Q BPDUs on VLAN 1 and tagged PVST+ BPDUs on other VLANs
- Tagged IEEE 802.1Q BPDUs on VLAN 1 and untagged BPDUs on another VLAN

Tagged Port Using Default VLAN 1 as its Port Native VLAN

Figure 9.29 shows an example of a PVST+ configuration that uses VLAN 1 as the untagged default VLAN and VLANs 2, 3, and 4 as tagged VLANs.

Figure 9.29 Default VLAN 1 for Untagged BPDUs


Untagged IEEE BPDU for VLAN 1
Untagged PVST BPDU for VLAN 1
Tagged PVST BPDUs for VLANs 2, 3, 4
Port 1/1
Port 3/2
Cisco device

To implement this configuration, enter the following commands.

Commands on the Foundry Device

```
FastIron(config)#vlan-group 1 vlan 2 to 4
FastIron(config-vlan-group-1)#tagged ethernet 1/1
FastIron(config-vlan-group-1)#exit
FastIron(config)#interface ethernet 1/1
FastIron(config-if-1/1)#dual-mode
FastIron(config-if-1/1)#pvst-mode
```

These commands configure a VLAN group containing VLANs 2, 3, and 4, add port 1/1 as a tagged port to the VLANs, and enable the dual-mode feature and PVST+ support on the port. The dual-mode feature allows the port to send and receive untagged frames for the default VLAN (VLAN 1 in this case) in addition to tagged frames for VLANs 2, 3, and 4. Enabling the PVST+ support ensures that the port is ready to send and receive PVST+ BPDUs. If you do not manually enable PVST+ support, the support is not enabled until the port receives a PVST+ BPDU.

The configuration leaves the default VLAN and the port's Port Native VLAN unchanged. The default VLAN is 1 and the port's Port Native VLAN also is 1. The dual-mode feature supports untagged frames on the default VLAN only. Thus, port 1/1 can send and receive untagged BPDUs for VLAN 1 and can send and receive tagged BPDUs for the other VLANs.

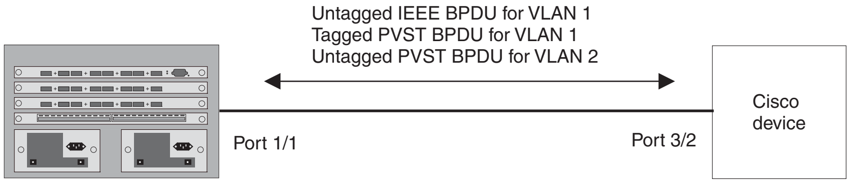
Port 1/1 will process BPDUs as follows:

- Process IEEE 802.1Q BPDUs for VLAN 1.
- Process tagged PVST BPDUs for VLANs 2, 3, and 4.
- Drop untagged PVST BPDUs for VLAN 1.

Untagged Port Using VLAN 2 as Port Native VLAN

Figure 9.30 shows an example in which a port's Port Native VLAN is not VLAN 1. In this case, VLAN 1 uses tagged frames and VLAN 2 uses untagged frames.

Figure 9.30 Port Native VLAN 2 for Untagged BPDUs


Untagged IEEE BPDU for VLAN 1
Tagged PVST BPDU for VLAN 1
Untagged PVST BPDU for VLAN 2
Port 1/1
Port 3/2
Cisco device

To implement this configuration, enter the following commands.

Commands on the Foundry Device

```
FastIron(config)#default-vlan-id 4000
FastIron(config)#vlan 1
FastIron(config-vlan-1)#tagged ethernet 1/1
FastIron(config-vlan-1)#exit
FastIron(config)#vlan 2
FastIron(config-vlan-2)#tagged ethernet 1/1
FastIron(config-vlan-2)#exit
FastIron(config)#interface ethernet 1/1
FastIron(config-if-1/1)#dual-mode 2
FastIron(config-if-1/1)#pvst-mode
FastIron(config-if-1/1)#exit
```

These commands change the default VLAN ID, configure port 1/1 as a tagged member of VLANs 1 and 2, and enable the dual-mode feature and PVST+ support on port 1/1. Since VLAN 1 is tagged in this configuration, the default VLAN ID must be changed from VLAN 1 to another VLAN ID. Changing the default VLAN ID from 1 allows the port to process tagged frames for VLAN 1. VLAN 2 is specified with the **dual-mode** command, which makes VLAN 2 the port's Port Native VLAN. As a result, the port processes untagged frames and untagged PVST BPDUs on VLAN 2.

NOTE: Although VLAN 2 becomes the port's untagged VLAN, the CLI still requires that you add the port to the VLAN as a tagged port, since the port is a member of more than one VLAN.

Port 1/1 will process BPDUs as follows:

- Process IEEE 802.1Q BPDUs for VLAN 1.
- Process untagged PVST BPDUs for VLAN 2.
- Drop tagged PVST BPDUs for VLAN 1.

Note that when VLAN 1 is not the default VLAN, the ports must have the dual-mode feature enabled in order to process IEEE 802.1Q BPDUs.

For example, the following configuration is incorrect:

```
FastIron(config)#default-vlan-id 1000
FastIron(config)#vlan 1
FastIron(config-vlan-1)#tagged ethernet 1/1 to 1/2
FastIron(config-vlan-1)#exit
FastIron(config)#interface ethernet 1/1
FastIron(config-if-1/1)#pvst-mode
FastIron(config-if-1/1)#exit
FastIron(config)#interface ethernet 1/2
FastIron(config-if-1/2)#pvst-mode
FastIron(config-if-1/2)#exit
```

In the configuration above, all PVST BPDUs associated with VLAN 1 would be discarded. Since IEEE BPDUs associated with VLAN 1 are untagged, they are discarded because the ports in VLAN 1 are tagged. Effectively, the BPDUs are never processed by the Spanning Tree Protocol. STP assumes that there is no better bridge on the network and sets the ports to FORWARDING. This could cause a Layer 2 loop.

The following configuration is correct:

```
FastIron(config)#default-vlan-id 1000
FastIron(config)#vlan 1
FastIron(config-vlan-1)#tagged ethernet 1/1 to 1/2
FastIron(config-vlan-1)#exit
FastIron(config)#interface ethernet 1/1
FastIron(config-if-1/1)#pvst-mode
FastIron(config-if-1/1)#dual-mode
FastIron(config-if-1/1)#exit
FastIron(config)#interface ethernet 1/2
FastIron(config-if-1/2)#pvst-mode
FastIron(config-if-1/2)#dual-mode
FastIron(config-if-1/2)#exit
```

Setting the ports as dual-mode ensures that the untagged IEEE 802.1Q BPDUs reach the VLAN 1 instance.

PVRST Compatibility

Platform Support:

- FESX/FSX/FWSX devices running software release 02.5.00 and later
- FGS and FLS devices running software release 02.5.00 and later

Support for Cisco's per-VLAN Rapid Spanning Tree (PVRST) is included with FastIron X Series and FastIron GS devices running software release 02.5.00 or later.

PVRST, the "rapid" version of per-VLAN spanning tree (PVST), is a Cisco proprietary protocol. PVRST corresponds to Foundry’s full implementation of IEEE 802.1w (RSTP). Likewise, PVST, also a Cisco proprietary protocol, corresponds to Foundry’s implementation of IEEE 802.1D (STP).

- In releases prior to 02.5.00, when a Foundry device receives a PVRST packet on a port running 802.1w, the 802.1w state machines for that port falls back to the classic or legacy, 802.1D compatibility mode. Once moved to the legacy 802.1D mode, the Foundry device works in the standard STP mode and the rapid convergence advantage of 802.1w is no longer available.
- Software release 02.5.00 implements compatibility with PVRST on Foundry’s FastIron X Series and FastIron GS devices. When a Foundry device receives PVRST BPDUs on a port configured to run 802.1w, it recognizes and processes these BPDUs and continues to operate in 802.1w mode.

PVRST compatibility is automatically enabled in software release 02.5.00 and later when a port receives a PVRST BPDU.

BPDU Guard

Platform Support:

- FESX/FSX/FWSX devices running software release 03.2.00 and later
- FGS and FLS devices running software release 03.0.00 and later

In an STP environment, switches, end stations, and other Layer 2 devices use Bridge Protocol Data Units (BPDUs) to exchange information that STP will use to determine the best path for data flow.

The BPDU guard, an enhancement to STP, removes a node that reflects BPDUs back in the network. It enforces the STP domain borders and keeps the active topology predictable by not allowing any network devices behind a BPDU guard-enabled port to participate in STP.

In some instances, it is unnecessary for a connected device, such as an end station, to initiate or participate in an STP topology change. In this case, you can enable the STP BPDU guard feature on the Foundry port to which the end station is connected. Foundry's STP BPDU guard shuts down the port and puts it into an errdisable state. This disables the connected device's ability to initiate or participate in an STP topology. A log message is then generated for a BPDU guard violation, and a CLI message is displayed to warn the network administrator of a severe invalid configuration. The BPDU guard feature provides a secure response to invalid configurations because the administrator must manually put the interface back in service if errdisable recovery is not enabled.

Enabling BPDU Protection by Port

You enable STP BPDU guard on individual interfaces. The feature is disabled by default.

To enable STP BPDU guard on a specific port, enter commands such as the following:

```
FastIron(config) interface ethe 2/1
FastIron(config-if-e1000-2/1)#stp-bpdu-guard
```

or

```
FGS624P Switch(config) interface ethe 0/1/2
FGS624P Switch(config-if-e1000-0/1/2)#stp-bpdu-guard
```

Syntax: [no] stp-bpdu-guard

The **no** parameter disables the BPDU guard on this interface.

You can also use the multiple interface command to enable this feature on multiple ports at once. For example,

```
FastIron(config)#interface ethernet 1/1 to 1/9
FastIron(config-mif-1/1-1/9)#stp-bpdu-guard
FastIron(config-mif-1/1-1/9)#
```

or

```
FGS624P Switch(config)#interface ethernet 0/1/1 to 0/1/9
FGS624P Switch(config-mif-0/1/1-0/1/9)#stp-bpdu-guard
FGS624P Switch(config-mif-0/1/1-0/1/9)#
```

This will enable stp-bpdu-guard on ports 0/1/1 to 0/1/9

Re-enabling Ports Disabled by BPDU Guard

When a BPDU Guard-enabled port is disabled by BPDU Guard, the Foundry device will place the port in ***errdisable*** state and display a message on the console indicating that the port is errdisabled (see “Example Console Messages”). In addition, the **show interface** command output will indicate that the port is errdisabled. For example:

```
FastIron#show int e 2
Gigabit Ethernet2 is ERR-DISABLED (bpduguard), line protocol is down
```

To re-enable a port that is in ***errdisable*** state, you must first disable the port then re-enable it. Enter commands such as the following:

```
FastIron(config)#int e 2
FastIron(config-if-e1000-2)#disable
FastIron(config-if-e1000-2)#enable
```

If you attempt to enable an errdisabled port without first disabling it, the following error message will appear on the console:

```
FastIron(config-if-e1000-2)#enable
Port 2 is errdisabled, do disable first and then enable to enable it
```

Displaying the BPDU Guard Status

To display the BPDU guard state, enter the **show running configuration** or the **show stp-bpdu-guard** command.

For FastIron X Series devices:

```
FastIron#show stp-bpdu-guard
BPDU Guard Enabled on:
Interface Violation
Port 1 No
Port 2 No
Port 3 No
Port 4 No
Port 5 No
Port 6 No
Port 7 No
Port 8 No
Port 9 No
Port 10 No
Port 11 No
Port 12 Yes
Port 13 No
```

For FGS devices:

```
FGS624P Switch#show stp-bpdu-guard
BPDU Guard Enabled on:
Ports: (Stk0/S1)   2   3   4   5   9  10  11  12  13  14  15  16
Ports: (Stk0/S1)  17  18  19  20  21  22  23  24
```

Syntax: show stp-bpdu-guard

Example Configurations

EXAMPLES:

The following example shows how to configure BPDU guard at interface level and to verify the configuration by issuing the **show stp-bpdu-guard** and the **show interface** commands.

For FESX/FSX/FWSX devices running software release 03.2.00:

```
FastIron Router(config)#interface ethernet 1
FastIron Router(config-if-e1000-1)#stp-bpdu-guard
FastIron Router(config-if-e1000-1)#
FastIron Router(config-if-e1000-1)#show stp-bpdu-guard
BPDU Guard Enabled on:
Port
1

FastIron(config-if-e1000-1)#
FastIron(config-if-e1000-1)#show interfaces ethernet 1
GigabitEthernet1 is up, line protocol is up
Hardware is GigabitEthernet, address is 000c.dba0.7100 (bia 000c.dba0.7100)
Configured speed auto, actual 100Mbit, configured duplex fdx, actual fdx
Configured mdi mode AUTO, actual MDI
Member of L2 VLAN ID 2, port is untagged, port state is FORWARDING
BPDU guard is Enabled, ROOT protect is Disabled
STP configured to ON, priority is level0, flow control enabled
mirror disabled, monitor disabled
Not member of any active trunks
Not member of any configured trunks
No port name
IPG MII 96 bits-time, IPG GMII 96 bits-time
IP MTU 1500 bytes
300 second input rate: 8 bits/sec, 0 packets/sec, 0.00% utilization
300 second output rate: 256 bits/sec, 0 packets/sec, 0.00% utilization
88 packets input, 15256 bytes, 0 no buffer
Received 75 broadcasts, 13 multicasts, 0 unicasts
1 input errors, 0 CRC, 0 frame, 0 ignored
0 runts, 0 giants
4799 packets output, 313268 bytes, 0 underruns
Transmitted 90 broadcasts, 4709
```

For FGS and FLS devices running software release 03.0.00:

```
FGS624P Switch(config)#interface ethernet 0/1/1
FGS624P Switch(config-if-e1000-0/1/1)#stp-bpdu-guard
FGS624P Switch(config-if-e1000-0/1/1)#exit

FGS624P Switch#show stp-bpdu-guard
```

```
BPDU Guard Enabled on:
Ports: (Stk0/S1)   1

FGS624P Switch#show interfaces ethernet 0/1/1
GigabitEthernet0/1/1 is down, line protocol is down
  Hardware is GigabitEthernet, address is 00e0.5204.4000 (bia 00e0.5204.4000)
  Configured speed auto, actual unknown, configured duplex fdx, actual unknown
  Configured mdi mode AUTO, actual unknown
  Member of L2 VLAN ID 1, port is untagged, port state is BLOCKING
  BPDU Guard is enabled, Root Protect is disabled
  STP configured to ON, priority is level0
  Flow Control is config enabled, oper disabled, negotiation disabled
  Mirror disabled, Monitor disabled
  Not member of any active trunks
  Not member of any configured trunks
  No port name
  Inter-Packet Gap (IPG) is 96 bit times
  300 second input rate: 0 bits/sec, 0 packets/sec, 0.00% utilization
  300 second output rate: 0 bits/sec, 0 packets/sec, 0.00% utilization
  0 packets input, 0 bytes, 0 no buffer
  Received 0 broadcasts, 0 multicasts, 0 unicasts
  0 input errors, 0 CRC, 0 frame, 0 ignored
  0 runts, 0 giants
  0 packets output, 0 bytes, 0 underruns
  Transmitted 0 broadcasts, 0 multicasts, 0 unicasts
  0 output errors, 0 collisions
FGS624P Switch(config)#
```

Example Console Messages

A console message such as the following is generated after a BPDU guard violation occurs on a system that is running MSTP.

```
FastIron(config-if-e1000-23)#MSTP: Received BPDU on BPDU guard enabled Port
23,errdisable Port 23
```

A console message such as the following is generated after a BPDU guard violation occurs on a system that is running STP.

```
FastIron(config)#STP: Received BPDU on BPDU guard enabled Port 23 (vlan=1),
errdisable Port 23
```

A console message such as the following is generated after a BPDU guard violation occurs on a system that is running RSTP.

```
FastIron(config-vlan-1)#RSTP: Received BPDU on BPDU guard enabled Port 23
(vlan=1),errdisable Port 23
```

Root Guard

Platform Support:

- FESX/FSX/FWSX devices running software release 03.2.00 and later
- FGS and FLS devices running software release 03.0.00 and later

The standard STP (802.1D), RSTP (802.1W) or 802.1S does not provide any way for a network administrator to securely enforce the topology of a switched layer 2 network. The forwarding topology of a switched network is calculated based on the root bridge position, along with other parameters. This means any switch can be the root bridge in a network as long as it has the lowest bridge ID. The administrator cannot enforce the position of the root bridge. A better forwarding topology comes with the requirement to place the root bridge at a specific predetermined location. Root Guard can be used to predetermine a root bridge location and prevent rogue or unwanted switches from becoming the root bridge.

When root guard is enabled on a port, it keeps the port in a designated role. If the port receives a superior STP Bridge Protocol Data Units (BPDU), it puts the port into a ROOT-INCONSISTANT state and triggers a log message and an SNMP trap. The ROOT-INCONSISTANT state is equivalent to the BLOCKING state in 802.1D and to the DISCARDING state in 802.1W. No further traffic is forwarded on this port. This allows the bridge to prevent traffic from being forwarded on ports connected to rogue or misconfigured STP bridges.

Once the port stops receiving superior BPDUs, root guard automatically sets the port back to learning, and eventually to a forwarding state through the spanning-tree algorithm.

Configure root guard on all ports where the root bridge should not appear. This establishes a protective network perimeter around the core bridged network, cutting it off from the user network.

NOTE: Root guard may prevent network connectivity if it is improperly configured. Root guard must be configured on the perimeter of the network rather than the core.

Root Guard is supported on the following:

- Devices running FSX software release 03.2.00 and later
- Devices running FGS software release 03.0.00 and later

NOTE: Root guard is not supported when MSTP is enabled.

Enabling STP Root Guard

An STP root guard is configured on an interface by entering commands similar to the following:

```
FastIron(config)#interface ethernet 5/5
```

```
FastIron(config-if-e10000-5/5)spanning-tree root-protect
```

Syntax: [no] spanning-tree root-protect

Enter the **no** form of the command to disable STP root guard on the port.

Displaying the STP Root Guard

To display the STP root guard state, enter the **show running configuration** or the **show spanning-tree root-protect** command.

```
FastIron#show spanning-tree root-protect
Root Protection Enabled on:
Port
1
```

Syntax: show spanning-tree root-protect

Displaying the Root Guard by VLAN

You can display root guard information for all VLANs or for a specific VLAN. For example, to display root guard violation information for VLAN 7.

Syntax: show spanning-tree [<vlan-id>]

If you do not specify a <vlan-id>, information for all VLANs is displayed. For example, to display root guard violation information for VLAN 7.

```
FastIron#show spanning-tree vlan 7
STP instance owned by VLAN 7
Global STP (IEEE 802.1D) Parameters:
VLAN Root Root Root Prio Max He- Ho- Fwd Last Chg Bridge
```

```
ID ID Cost Port rity Age llo ld dly Chang cnt Address
Hex sec sec sec sec sec
7 a000000011112220 0 Root a000 20 2 1 15 4 4 000011112220
Port STP Parameters:
Port Prio Path State Fwd Design Designated Designated
Num rity Cost Trans Cost Root Bridge
Hex
1 80 19 ROOT-INCONS 2 0 a000000011112220 a000000011112220
```

Error Disable Recovery

In case a BPDU guard violation occurs, a port is placed into an errdisable state which is functionally equivalent to a Disable state. Once in an errdiable state, it remains in that state until one of the following methods is used to return the port to an Enabled state:

1. Manually disabling and enabling that interface
2. Automatically, through the errdisable recovery mechanism

The **errdisable recovery interval** command is used to configure a time-out for ports in errdisable state, after which the ports are re-enabled automatically.

When BPDU guard puts a port into errdisabled state, the port remains in errdisabled state unless it is enabled manually by issuing a **disable** command and then the **enable** command on the associated interface or you have errdisable recovery turned on. The **errdisable** command allows you to choose the type of error that automatically reenables the port after a specified amount of time.

Enabling Error Disable Recovery

To enable errdisable recovery for BPDU guard, enter a command such as the following:

```
FastIron(config)#errdisable recovery cause bpduguard
```

To enable error disable recovery for any reason, enter a command such as the following:

```
FastIron(config)#errdisable recovery cause all
```

Syntax: errdisable recovery [cause < bpduguard | all >]

The **cause** is the reason why the port is in the errdisable state. Valid values are **bpdu-guard** and **all**.

Use the **bpduguard** parameter to allow the port to recover from the errdisabled state, if the state was caused by a BPDU guard violation.

The **all** parameter allows ports to recover from an errdisabled state, if the state was caused by any reason other than a BPDU guard violation.

Setting the Recovery Interval

The **errdisable recovery interval** command allows you to configure a timeout for ports in errdisable state, after which the ports are reenabled automatically. To set the errdisable recovery time-out interval, enter a command such as the following:

```
FastIron(config)#errdisable recovery interval 20
```

Syntax: [no] errdisable recovery interval <seconds>

The **seconds** parameter allows you to set the timeout value for the recovery mechanism when the port is in an errdisabled state. Once this timeout value expires, the ports are automatically re-enabled. Valid values are from 10 to 65535 seconds (10 seconds to 24 hours).

Displaying the Error Disable Recovery State by Interface

The port status of errdisabled displays in the output of the **show interface** and the **show interface brief** commands. In this example, errdisable is enabled on interface ethernet 1 and errdisable is enabled because of a BPDU guard violation.

```
FastIron#show interfaces ethernet 1
GigabitEthernet1 is ERR-DISABLED (bpduguard), line protocol is down
BPDU guard is Enabled, ROOT protect is Disabled
Hardware is GigabitEthernet, address is 000c.dba0.7100 (bia 000c.dba0.7100)
Configured speed auto, actual unknown, configured duplex fdx, actual unknown
Configured mdi mode AUTO, actual unknown
Member of L2 VLAN ID 2, port is untagged, port state is DISABLED
STP configured to ON, priority is level0, flow control enabled
mirror disabled, monitor disabled
Not member of any active trunks
Not member of any configured trunks
No port name
IPG MII 96 bits-time, IPG GMII 96 bits-time
IP MTU 1500 bytes
300 second input rate: 0 bits/sec, 0 packets/sec, 0.00% utilization
300 second output rate: 0 bits/sec, 0 packets/sec, 0.00% utilization
145 packets input, 23561 bytes, 0 no buffer
Received 124 broadcasts, 21 multicasts, 0 unicasts
1 input errors, 0 CRC, 0 frame, 0 ignored
0 runts, 0 giants
5067 packets output, 330420 bytes, 0 underruns
Transmitted 90 broadcasts, 4977 multicasts, 0 unicasts
0 output errors, 0 collisions
```

Displaying the Recovery State for All Conditions

Use the **show errdisable recovery** command to display all the default error disable recovery state for all possible conditions. In this example, port 6 is undergoing a recovery.

```
FastIron#show errdisable recovery

ErrDisable Reason Timer Status
--------------------------------------
all reason Disabled
bpduguard Enabled
Timeout Value: 300 seconds
Interface that will be enabled at the next timeout:
Interface Errdisable reason Time left (sec)
-------------- ----------------- ---------------
Port 6 bpduguard 297
```

Syntax: show errdisable recovery

Displaying the Recovery State by Port Number and Cause

To see which ports are under an errdisabled state, use the **show errdisable summary** command. This command not only shows the port number, but also displays the reason why the port is in an errdisable state and the method used to recover the port. In this example, port 6 is errdisabled for a BPDU guard violation.

```
FastIron#show errdisable summary
Port 6 ERR_DISABLED for bpduguard
```

Syntax: show errdisable summary

Errdisable Syslog Messages

When the system places a port into an errdisabled state for BPDU guard, a log message is generated. When the errdisable recovery timer expires, a log message is also generated.

A Syslog message such as the following is generated after a port is placed into an errdisable state for BPDU guard.

```
STP: VLAN 50 BPDU-guard port 3 detect (Received BPDU), putting into err-disable
state
```

A Syslog message such as the following is generated after the recovery timer expires.

```
ERR_DISABLE: Interface ethernet 3, err-disable recovery timeout
```

802.1s Multiple Spanning Tree Protocol

Platform Support:

- FESX/FSX/FWSX devices running software release 03.2.00 and later
- FGS and FLS devices running software release 04.0.00 and later

Multiple Spanning Tree Protocol (MSTP), as defined in IEEE 802.1s, allows multiple VLANs to be managed by a single STP instance and supports per-VLAN STP. As a result, several VLANs can be mapped to a reduced number of spanning-tree instances. This ensures loop-free topology for one or more VLANs that have the similar layer-2 topology. The Foundry implementation supports up to 16 spanning tree instances in an MSTP enabled bridge which means that it can support up to 16 different Layer 2 topologies. The spanning tree algorithm used by MSTP is RSTP which provides quick convergence.

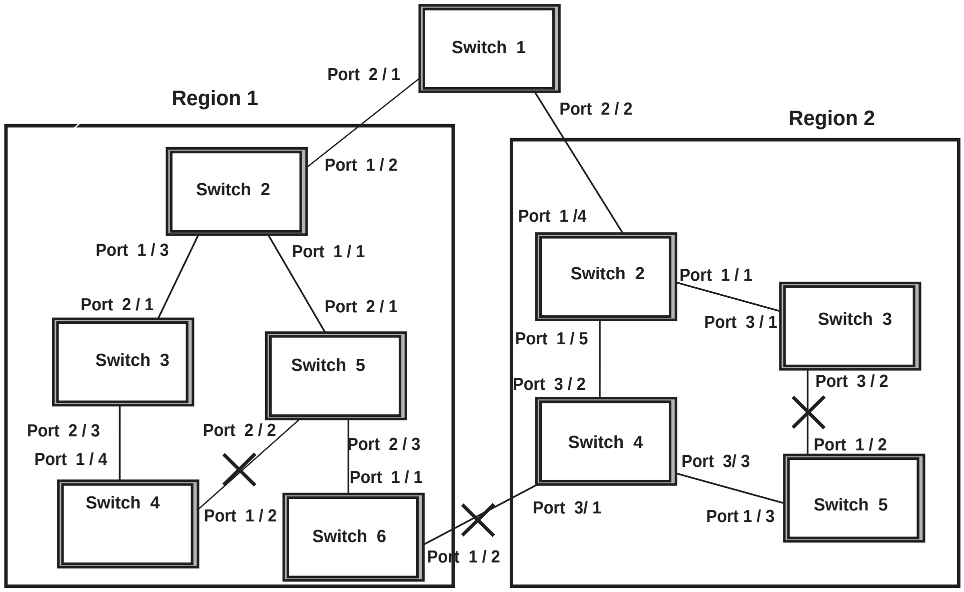
Multiple Spanning-Tree Regions

Using MSTP, the entire network runs a common instance of RSTP. Within that common instance, one or more VLANs can be individually configured into distinct regions. The entire network runs the common spanning tree instance (CST) and the regions run a local instance. The local instance is known as Internal Spanning Tree (IST). The CST treats each instance of IST as a single bridge. Consequently, ports are blocked to prevent loops that might occur within an IST and also throughout the CST. With the exception of the provisions for multiple instances, MSTP operates exactly like RSTP.

For example, in Figure 9.31 a network is configured with two regions: Region 1 and Region 2. The entire network is running an instance of CST. Each of the regions is running an instance of IST. In addition, this network contains Switch 1 running MSTP that isn’t configured in a region and consequently is running in the CIST instance. In this configuration, the regions are each regarded as a single bridge to the rest of the network, as is Switch 1. The CST prevents loops from occurring across the network. Consequently, a port is blocked at port 1/2 of switch 4.

Additionally, loops must be prevented in each of the IST instances. Within the IST Region 1, a port is blocked at port 1/2 of switch 4 to prevent a loop in that region. Within Region 2, a port is blocked at port 3/2 of switch 3 to prevent a loop in that region.

Figure 9.31 MSTP Configured Network



The following definitions describe the STP instances that define an MSTP configuration:

Common Spanning (CST) – CST is defined in 802.1q and assumes one spanning-tree instance for the entire bridged network regardless of the number of VLANs. In MSTP, an MSTP region appears as a virtual bridge that runs CST.

Internal Spanning Tree (IST) – IST is a new terminology introduced in 802.1s. An MSTP bridge must handle at least these two instances: one IST and one or more MSTIs (Multiple Spanning Tree Instances). Within each MST region, the MSTP maintains multiple spanning-tree instances. Instance 0 is a special instance known as IST, which extends CST inside the MST region. IST always exists if the switch runs MSTP. Besides IST, this implementation supports up to 15 MSTIs, numbered from 1 to 4094.

Common and Internal Spanning Trees (CIST) – CIST is a collection of the ISTs in each MST region and the CST that interconnects the MST regions and single spanning trees.

Multiple Spanning Tree Instance (MSTI) – The MSTI is identified by an MST identifier (MSTid) value between 1 and 4094.

MSTP Region – These are clusters of bridges that run multiple instances of the MSTP protocol. Multiple bridges detect that they are in the same region by exchanging their configuration (instance to VLAN mapping), name, and revision-level. Therefore, if you need to have two bridges in the same region, the two bridges must have identical configurations, names, and revision-levels. Also, one or more VLANs can be mapped to one MSTP instance (IST or MSTI) but a VLAN cannot be mapped to multiple MSTP instances.

NOTE: One or more VLANs can be mapped to one MSTP instance (IST or MSTI) but a VLAN cannot be mapped to multiple MSTP instances.

Configuration Notes

When configuring MSTP, note the following:

- With MSTP running, enabling static trunk on ports that are members of many VLANs (4000 or more VLANs) will keep the system busy for 20 to 25 seconds.

Configuring MSTP Mode and Scope

With the introduction of MSTP, a system can be either under MSTP mode or not under MSTP mode. The default state is to **not** be under MSTP mode. MSTP configuration can only be performed in a system under MSTP mode.

With a system configured under MSTP mode, there is a concept called MSTP scope. MSTP scope defines the VLANs that are under direct MSTP control. You cannot run 802.1D or 802.1w on any VLAN (even outside of MSTP scope) and you cannot create topology groups when a system is under MSTP mode. While a VLAN group will still be supported when a system is under MSTP mode, the member VLAN should either be all in the MSTP scope or all out of the MSTP scope.

When a system is configured from non-MSTP mode to MSTP mode, the following changes are made to the system's configuration:

- All 802.1D and 802.1w STP instances are deleted regardless of whether the VLAN is inside the MSTP scope or not
- All topology groups are deleted
- Any GVRP configuration is deleted
- Any VSRP configuration is deleted
- Single-span (if configured) is deleted
- MRP running on a VLAN inside MSTP scope is deleted
- The CIST is created and all VLANS inside the MSTP scope are attached with the CIST

Make sure that no physical layer-2 loops exist prior to switching from non-MSTP mode to MSTP mode. If, for example, you have an L2 loop topology configured as a redundancy mechanism before you perform the switch, a Layer 2 storm should be expected.

To configure a system into MSTP mode, use the following command at the Global Configuration level:

```
FastIron(config)#mstp scope all
```

Syntax: [no] mstp scope all

NOTE: MSTP is not operational however until the **mstp start** command is issued as described in "Activating MSTP on a Switch" on page 9-80.

Once the system is configured into MSTP mode, CIST (sometimes referred to as "instance 0") is created and all existing VLANs inside the MSTP scope are controlled by CIST. In addition, whenever you create a new VLAN inside MSTP scope, it is put under CIST control by default. In the Foundry MSTP implementation however, a VLAN ID can be pre-mapped to another MSTI as described in "Configuring an MSTP Instance" on page 9-78. A VLAN whose ID is pre-mapped, will attach to the specified MSTI instead of to the CIST when created.

NOTE: Once under MSTP mode, CIST always controls all ports in the system. If you do not want a port to run MSTP, configure the **no spanning-tree** command under the specified interface configuration.

Using the **[no]** option on a system that is configured for MSTP mode changes the system to non-MSTP mode. When this switch is made, all MSTP instances are deleted together with all MSTP configurations. ALL VLANs inside the original MSTP scope will not run any Layer-2 protocols after the switch.

Configuring Additional MSTP Parameters

To configure a switch for MSTP, you could configure the name and the revision on each switch that is being configured for MSTP. You must then create an MSTP Instance and assign an ID. VLANs are then assigned to MSTP instances. These instances must be configured on all switches that interoperate with the same VLAN assignments. Port cost, priority and global parameters can then be configured for individual ports and instances. In addition, operational edge ports and point-to-point links can be created and MSTP can be disabled on individual ports.

Each of the commands used to configure and operate MSTP are described in the following:

- "Setting the MSTP Name"

- “Setting the MSTP Revision Number”
- “Configuring an MSTP Instance”
- “Configuring Bridge Priority for an MSTP Instance”
- “Setting the MSTP Global Parameters”
- “Setting Ports To Be Operational Edge Ports”
- “Setting Automatic Operational Edge Ports”
- “Setting Point-to-Point Link”
- “Disabling MSTP on a Port”
- “Forcing Ports to Transmit an MSTP BPDU”
- “Activating MSTP on a Switch”

Setting the MSTP Name

Each switch that is running MSTP is configured with a name. It applies to the switch which can have many different VLANs that can belong to many different MSTP regions.

To configure an MSTP name, use a command such as the following at the Global Configuration level:

```
FastIron(config)#mstp name foundry
```

Syntax: [no] mstp name <name>

The **name** parameter defines an ASCII name for the MSTP configuration. The default name is for the name variable to be blank.

Setting the MSTP Revision Number

Each switch that is running MSTP is configured with a revision number. It applies to the switch which can have many different VLANs that can belong to many different MSTP regions.

To configure an MSTP revision number, use a command such as the following at the Global Configuration level:

```
FastIron(config)#mstp revision 4
```

Syntax: [no] mstp revision <revision-number>

The **revision** parameter specifies the revision level for MSTP that you are configuring on the switch. It can be a number from 0 and 65535. The default revision number is 0.

Configuring an MSTP Instance

An MSTP instance is configured with an MSTP ID for each region. Each region can contain one or more VLANs. Foundry’s implementation of MSTP allows you to assign VLANS or ranges of VLANs to an MSTP instance before or after they have been defined. If pre-defined, a VLAN will be placed in the MSTI that it was assigned to immediately when the VLAN is created. Otherwise, the default operation is to condition of assign all new VLANs to the CIST. VLANs assigned to the CIST by default can be moved later to a specified MSTI.

To configure an MSTP instance and map one or more VLANs to that MSTI, use a command such as the following at the Global Configuration level:

```
FastIron(config) #mstp instance 7 vlan 4 to 7
```

Syntax: [no] mstp instance <instance-number> [vlan <vlan-id> | vlan-group <group-id>]

The **instance** parameter defines the number for the instance of MSTP that you are configuring. The value 0 (which identifies the CIST) cannot be used. You can have up to 15 instances, number 1 – 4094.

The **vlan** parameter assigns one or more VLANs or a range of VLANs to the instance defined in this command.

The **vlan-group** parameter assigns one or more VLAN groups to the instance defined in this command.

The **no** option moves a VLAN or VLAN group from it’s assigned MSTI back into the CIST.

NOTE: The system does not allow an MSTI without any VLANs mapped to it. Consequently, removing all VLANs from an MSTI, deletes the MSTI from the system. The CIST by contrast will exist regardless of whether or not any VLANs are assigned to it or not. Consequently, if all VLANs are moved out of a CIST, the CIST will still exist and functional.

Configuring Bridge Priority for an MSTP Instance

Priority can be configured for a specified instance. To configure priority for an MSTP instance, use a command such as the following at the Global Configuration level:

```
FastIron(config)#mstp instance 1 priority 8192
```

Syntax: [no] mstp instance <instance-number> priority <priority-value>

The <instance-number> variable is the number for the instance of MSTP that you are configuring.

You can set a **priority** to the instance that gives it forwarding preference over lower priority instances within a VLAN or on the switch. A higher number for the priority variable means a lower forwarding priority. Acceptable values are 0 - 61440 in increments of 4096. The default value is 32768.

Setting the MSTP Global Parameters

MSTP has many of the options available in RSTP as well as some unique options. To configure MSTP Global parameters for all instances on a switch:

```
FastIron(config)#mstp force-version 0 forward-delay 10 hello-time 4 max-age 12 max-
hops 9
```

Syntax: [no] mstp force-version <mode-number> forward-delay <value> hello-time <value> max-age <value> max-hops <value>

The **force-version** parameter forces the bridge to send BPDUs in a specific format. You can specify one of the following <mode-number> values:

- 0 – The STP compatibility mode. Only STP BPDUs will be sent. This is equivalent to single STP.
- 2 – The RSTP compatibility mode. Only RSTP BPDUS will be sent. This is equivalent to single STP.
- 3 – MSTP mode. In this default mode, only MSTP BPDUS will be sent.

The **forward-delay** <value> specifies how long a port waits before it forwards an RST BPDU after a topology change. This can be a value from 4 – 30 seconds. The default is 15 seconds.

The **hello-time** <value> parameter specifies the interval between two hello packets. The parameter can have a value from 1 – 10 seconds. The default is 2 seconds.

The **max-age** <value> parameter specifies the amount of time the device waits to receive a hello packet before it initiates a topology change. You can specify a value from 6 – 40 seconds, where the value adheres to the following formula:

max age $\geq$ 2 x (hello-time + 1) AND max age $\leq$ 2 x (forward-delay – 1)

The default max-age is 20 seconds.

The **max-hops** <value> parameter specifies the maximum hop count. You can specify a value from 1 – 40 hops. The default value is 20 hops.

Setting Ports To Be Operational Edge Ports

You can define specific ports as edge ports for the region in which they are configured to connect to devices (such as a host) that are not running STP, RSTP, or MSTP. If a port is connected to an end device such as a PC, the port can be configured as an edge port. To configure ports as operational edge ports enter a command such as the following:

```
FastIron(config)#mstp admin-edge-port ethernet 3/1
```

Syntax: [no] mstp admin-edge-port ethernet <slot/port>

The <slot/port> parameter specifies a port or range of ports as edge ports in the instance they are configured in.

Setting Automatic Operational Edge Ports

You can configure a FastIron router to automatically set a port as an operational edge port if the port does not receive any BPDUs since link-up. If the port receives a BPDU later, it is automatically reset to become an operational non-edge port. This feature is set globally to apply to all ports on a router where it is configured. This feature is configured as shown in the following:

```
FastIron(config)#mstp edge-port-auto-detect
```

Syntax: [no] mstp edge-port-auto-detect

NOTE: If this feature is enabled, it takes the port about 3 seconds longer to come to the enable state.

Setting Point-to-Point Link

You can set a point-to-point link between ports to increase the speed of convergence. To create a point-to-point link between ports, use a command such as the following at the Global Configuration level:

```
FastIron(config)#mstp admin-pt2pt-mac ethernet 2/5 ethernet 4/5
```

Syntax: [no] mstp admin-pt2pt-mac ethernet <slot/port>

The <slot/port> parameter specifies a port or range of ports to be configured for point-to-point links to increase the speed of convergence.

Disabling MSTP on a Port

To disable MSTP on a specific port, use a command such as the following at the Global Configuration level:

```
FastIron(config)#mstp disable ethernet 2/1
```

Syntax: [no] mstp disable ethernet <slot/port>

The <slot/port> variable specifies the location of the port that you want to disable MSTP for.

NOTE: When a port is disabled for MSTP, it behaves as blocking for all the VLAN traffic that is controlled by MSTIs and the CIST.

Forcing Ports to Transmit an MSTP BPDU

To force a port to transmit an MSTP BPDU, use a command such as the following at the Global Configuration level:

```
FastIron(config)#mstp force-migration-check ethernet 3/1
```

Syntax: [no] mstp force-migration-check ethernet <slot/port>

The <slot/port> variable specifies the port or ports that you want to transmit an MSTP BPDU from.

Activating MSTP on a Switch

MSTP scope must be enabled on the switch as described in “Configuring MSTP Mode and Scope” on page 9-76 before MSTP can be enabled.

To enable MSTP on your switch, use the following at the Global Configuration level:

```
FastIron(config)#mstp start
```

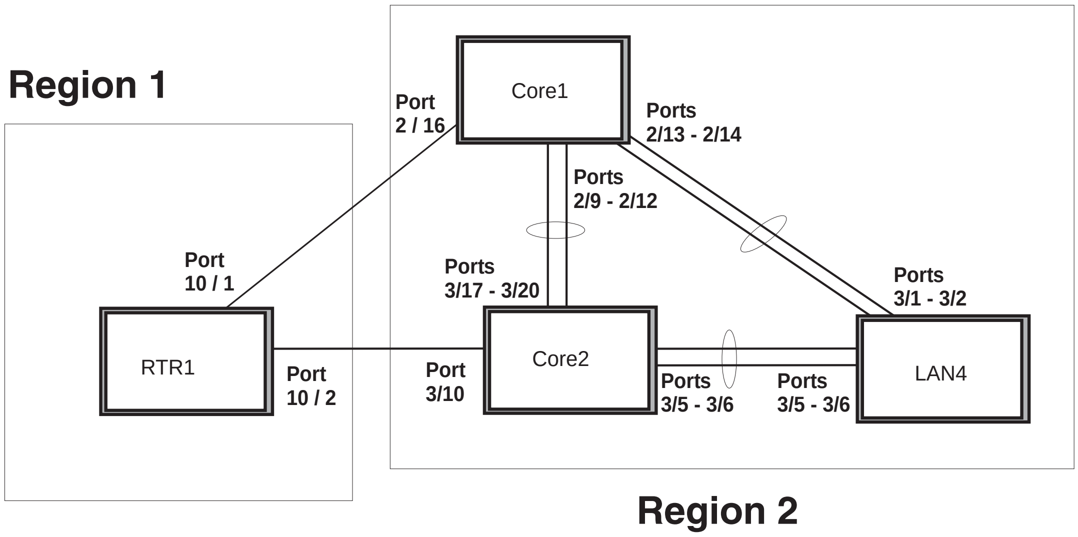
Syntax: [no] mstp start

The **[no]** option disables MSTP from operating on a switch.

EXAMPLES:

In Figure 9.32 four Foundry device routers are configured in two regions. There are four VLANs in four instances in Region 2. Region 1 is in the CIST.

Figure 9.32 **SAMPLE MSTP Configuration**



RTR1 Configuration

```
FastIron(config-vlan-4093)#tagged ethernet 10/1 to 10/2
FastIron(config-vlan-4093)#exit
FastIron(config)#mstp scope all
FastIron(config)#mstp name Reg1
FastIron(config)#mstp revision 1
FastIron(config)#mstp admin-pt2pt-mac ethernet 10/1 to 10/2
FastIron(config)#mstp start
FastIron(config)#hostname RTR1
```

Core 1 Configuration

```
FastIron(config)#trunk ethernet 2/9 to 2/12 ethernet 2/13 to 2/14
FastIron(config-vlan-1)#name DEFAULT-VLAN by port
FastIron(config-vlan-1)#exit
FastIron(config)#vlan 20 by port
FastIron(config-vlan-20)#tagged ethernet 2/9 to 2/14 ethernet 2/16
FastIron(config-vlan-20)#exit
FastIron(config)#vlan 21 by port
FastIron(config-vlan-21)#tagged ethernet 2/9 to 2/14 ethernet 2/16
FastIron(config-vlan-21)#exit
FastIron(config)#vlan 22 by port
FastIron(config-vlan-22)#tagged ethernet 2/9 to 2/14 ethernet 2/16
FastIron(config-vlan-22)#exit
FastIron(config)#vlan 23 by port
FastIron(config)#mstp scope all
FastIron(config)#mstp name HR
FastIron(config)#mstp revision 2
FastIron(config)#mstp instance 20 vlan 20
FastIron(config)#mstp instance 21 vlan 21
FastIron(config)#mstp instance 22 vlan 22
FastIron(config)#mstp instance 0 priority 8192
FastIron(config)#mstp admin-pt2pt-mac ethernet 2/9 to 2/14
FastIron(config)#mstp admin-pt2pt-mac ethernet 2/16
FastIron(config)#mstp disable ethernet 2/240.
FastIron(config)#mstp start
FastIron(config)#hostname CORE1
```

Core2 Configuration

```
FastIron(config)#trunk ethernet 3/5 to 3/6 ethernet 3/17 to 3/20
FastIron(config)#vlan 1 name DEFAULT-VLAN by port
FastIron(config-vlan-1)#exit
FastIron(config)#vlan 20 by port
FastIron(config-vlan-20)#tagged ethernet 3/5 to 3/6 ethernet 3/17 to 3/20
FastIron(config-vlan-20)#exit
FastIron(config)#vlan 21 by port
FastIron(config-vlan-21)#tagged ethernet 3/5 to 3/6 ethernet 3/17 to 3/20
FastIron(config-vlan-21)#exit
FastIron(config)#vlan 22 by port
FastIron(config-vlan-22)#tagged ethernet 3/5 to 3/6 ethernet 3/17 to 3/20
FastIron(config-vlan-22)#exit
FastIron(config)#mstp scope all
FastIron(config)#mstp name HR
FastIron(config)#mstp revision 2
FastIron(config)#mstp instance 20 vlan 20
FastIron(config)#mstp instance 21 vlan 21
FastIron(config)#mstp instance 22 vlan 22
FastIron(config)#mstp admin-pt2pt-mac ethernet 3/17 to 3/20 ethernet 3/5 to 3/6
FastIron(config)#mstp admin-pt2pt-mac ethernet 3/10
FastIron(config)#mstp disable ethernet 3/7 ethernet 3/24
FastIron(config)#mstp start
FastIron(config)#hostname CORE2
```

LAN 4 Configuration

```
FastIron(config)#trunk ethernet 3/5 to 3/6 ethernet 3/1 to 3/2
FastIron(config)#vlan 1 name DEFAULT-VLAN by port
FastIron(config-vlan-1)#exit
FastIron(config)#vlan 20 by port
FastIron(config-vlan-20)#tagged ethernet 3/1 to 3/2 ethernet 3/5 to 3/6
FastIron(config)#exit
FastIron(config)#vlan 21 by port
FastIron(config-vlan-21)#tagged ethernet 3/1 to 3/2 ethernet 3/5 to 3/6
FastIron(config-vlan-21)#exit
FastIron(config)#vlan 22 by port
FastIron(config-vlan-22)#tagged ethernet 3/1 to 3/2 ethernet 3/5 to 3/6
FastIron(config)#mstp scope all
FastIron(config)#mstp config name HR
FastIron(config)#mstp revision 2
FastIron(config)#mstp instance 20 vlan 20
FastIron(config)#mstp instance 21 vlan 21
FastIron(config)#mstp instance 22 vlan 22
FastIron(config)#mstp admin-pt2pt-mac ethernet 3/5 to 3/6 ethernet 3/1 to 3/2
FastIron(config)#mstp start
FastIron(config)#hostname LAN4
```

Displaying MSTP Statistics

MSTP statistics can be displayed using the commands shown below.

To display all general MSTP information, enter the following command:

```
FastIron#show mstp
MSTP Instance 0 (CIST) - VLANs: 1
----------------------------------------------------------------------------
Bridge           Bridge Bridge Bridge Bridge Root   Root  Root   Root
Identifier       MaxAge Hello  FwdDly Hop    MaxAge Hello FwdDly Hop
hex              sec    sec    sec    cnt    sec    sec   sec    cnt
8000000cdb80af01 20     2      15     20     20     2     15     19

Root             ExtPath   RegionalRoot     IntPath   Designated       Root
Bridge           Cost      Bridge           Cost      Bridge           Port
hex                        hex                        hex
8000000480bb9876 2000      8000000cdb80af01 0         8000000480bb9876 3/1

Port  Pri PortPath  P2P Edge Role       State      Designa-  Designated
Num       Cost      Mac Port                       ted cost  bridge
3/1   128 2000      T   F    ROOT       FORWARDING 0         8000000480bb9876

MSTP Instance 1 - VLANs: 2
----------------------------------------------------------------------------
Bridge           Max RegionalRoot     IntPath   Designated       Root  Root
Identifier       Hop Bridge           Cost      Bridge           Port  Hop
hex              cnt hex                        hex                    cnt
8001000cdb80af01 20  8001000cdb80af01 0         8001000cdb80af01 Root  20

Port  Pri PortPath  Role       State      Designa-  Designated
Num       Cost                            ted cost  bridge
3/1   128 2000      MASTER     FORWARDING 0         8001000cdb80af01
```

Syntax: show mstp <instance-number>

The <instance-number> variable specifies the MSTP instance that you want to display information for.

Table 9.10: Output from Show MSTP

This Field...	Displays...
MSTP Instance	The ID of the MSTP instance whose statistics are being displayed. For the CIST, this number is 0.
VLANs:	The number of VLANs that are included in this instance of MSTP. For the CIST this number will always be 1.
Bridge Identifier	The MAC address of the bridge.
Bridge MaxAge sec	Displays configured Max Age.
Bridge Hello sec	Displays configured Hello variable.
Bridge FwdDly sec	Displays configured FwdDly variable.
Bridge Hop cnt	Displays configured Max Hop count variable.
Root MaxAge sec	Max Age configured on the root bridge.
Root Hello sec	Hello interval configured on the root bridge.

Table 9.10: Output from Show MSTP (Continued)

This Field...	Displays...
Root FwdDly sec	FwdDly interval configured on the root bridge.
Root Hop Cnt	Current hop count from the root bridge.
Root Bridge	Bridge identifier of the root bridge.
ExtPath Cost	The configured path cost on a link connected to this port to an external MSTP region.
Regional Root Bridge	The Regional Root Bridge is the MAC address of the Root Bridge for the local region.
IntPath Cost	The configured path cost on a link connected to this port within the internal MSTP region.
Designated Bridge	The MAC address of the bridge that sent the best BPDU that was received on this port.
Root Port	Port indicating shortest path to root. Set to "Root" if this bridge is the root bridge.
Port Num	The port number of the interface.
Pri	The configured priority of the port. The default is 128.
PortPath Cost	Configured or auto detected path cost for port.
P2P Mac	Indicates if the port is configured with a point-to-point link: • **T** – The port is configured in a point-to-point link • **F** – The port is not configured in a point-to-point link
Edge	Indicates if the port is configured as an operational edge port: • **T** – indicates that the port is defined as an edge port. • **F** – indicates that the port is not defined as an edge port
Role	The current role of the port: • Master • Root • Designated • Alternate • Backup • Disabled
State	The port’s current spanning tree state. A port can have one of the following states: • Forwarding • Discarding • Learning • Disabled
Designated Cost	Port path cost to the root bridge.

Table 9.10: Output from Show MSTP (Continued)

This Field...	Displays...
Max Hop cnt	The maximum hop count configured for this instance.
Root Hop cnt	Hop count from the root bridge.

Displaying MSTP Information for a Specified Instance

The following example displays MSTP information specified for an MSTP instance.

```
FastIron#show mstp 1

MSTP Instance 1 - VLANs: 2
----------------------------------------------------------------------------
Bridge           Max RegionalRoot     IntPath   Designated       Root  Root
Identifier       Hop Bridge           Cost      Bridge           Port  Hop
hex              cnt hex                        hex                    cnt
8001000cdb80af01 20  8001000cdb80af01 0         8001000cdb80af01 Root  20

Port  Pri PortPath  Role       State      Designa-  Designated
Num       Cost                            ted cost  bridge
3/1   128 2000      MASTER     FORWARDING 0         8001000cdb80af01
```

See Table 9.10 for details about the display parameters.

Displaying MSTP Information for CIST Instance 0

Instance 0 is the Common and Internal Spanning Tree Instance (CIST). When you display information for this instance there are some differences with displaying other instances. The following example displays MSTP information for CIST Instance 0.

```
FastIron#show mstp 0

MSTP Instance 0 (CIST) - VLANs: 1
----------------------------------------------------------------------------
Bridge           Bridge Bridge Bridge Bridge Root   Root  Root   Root
Identifier       MaxAge Hello  FwdDly Hop    MaxAge Hello FwdDly Hop
hex              sec    sec    sec    cnt    sec    sec   sec    cnt
8000000cdb80af01 20     2      15     20     20     2     15     19

Root             ExtPath   RegionalRoot     IntPath   Designated       Root
Bridge           Cost      Bridge           Cost      Bridge           Port
hex                        hex                        hex
8000000480bb9876 2000      8000000cdb80af01 0         8000000480bb9876 3/1

Port  Pri PortPath  P2P Edge Role       State      Designa-  Designated
Num       Cost      Mac Port                       ted cost  bridge
3/1   128 2000      T   F    ROOT       FORWARDING 0         8000000480bb9876
```

To display details about the MSTP configuration, enter the following command:

```
FastIron#show mstp conf

MSTP CONFIGURATION
------------------
Name     : Reg1
Revision : 1
Version  : 3 (MSTP mode)
Status   : Started

Instance VLANs
-------- ------------------------------------------------------
0        4093
```

To display details about the MSTP that is configured on the device, enter the following command:

```
FastIron#show mstp detail
MSTP Instance 0 (CIST) - VLANs: 4093
----------------------------------------------------------------------------
Bridge: 800000b000c00000 [Priority 32768, SysId 0, Mac 00b000c00000]
FwdDelay 15, HelloTime 2, MaxHops 20, TxHoldCount 6

Port 6/54 - Role: DESIGNATED - State: FORWARDING
PathCost 20000, Priority 128, OperEdge T, OperPt2PtMac F, Boundary T
Designated - Root 800000b000c00000, RegionalRoot 800000b000c00000,
Bridge 800000b000c00000, ExtCost 0, IntCost 0
ActiveTimers - helloWhen 1
MachineState - PRX-DISCARD, PTX-IDLE, PPM-SENDING_RSTP, PIM-CURRENT
PRT-ACTIVE_PORT, PST-FORWARDING, TCM-INACTIVE
BPDUs - Rcvd MST 0, RST 0, Config 0, TCN 0
Sent MST 6, RST 0, Config 0, TCN 0
```

See Table 9.10 for explanation about the parameters in the output.

Syntax: show mstp [<mstp-id> | configuration | detail] [| begin <string> | exclude <string> | include <string>]

Enter an MSTP ID for <mstp-id>.

Chapter 10

Configuring Metro Features

This chapter describes how to configure Metro features. You can use Metro features individually or in combination to provide fast, reliable, and easy to configure Layer 2 connectivity in your Metro network.

Topology Groups

A topology group is a named set of VLANs that share a Layer 2 topology. Topology groups simplify configuration and enhance scalability of Layer 2 protocols by allowing you to run a single instance of a Layer 2 protocol on multiple VLANs.

You can use topology groups with the following Layer 2 protocols:

- STP
- MRP
- VSRP
- 802.1W

Topology groups simplify Layer 2 configuration and provide scalability by enabling you to use the same instance of a Layer 2 protocol for multiple VLANs. For example, if a Foundry device is deployed in a Metro network and provides forwarding for two MRP rings that each contain 128 VLANs, you can configure a topology group for each ring. If a link failure in a ring causes a topology change, the change is applied to all the VLANs in the ring's topology group. Without topology groups, you would need to configure a separate ring for each VLAN.

NOTE: If you plan to use a configuration saved under an earlier software release and the configuration contains STP groups, the CLI converts the STP groups into topology groups when you save the configuration. For backward compatibility, you can still use the STP group commands. However, the CLI converts the commands into the topology group syntax. Likewise, the **show stp-group** command displays STP topology groups.

Master VLAN and Member VLANs

Each topology group contains a master VLAN and can contain one or more member VLANs and VLAN groups.

- ***Master VLAN*** – The master VLAN contains the configuration information for the Layer 2 protocol. For example, if you plan to use the topology group for MRP, the topology group's master VLAN contains the ring configuration information.
- ***Member VLANs*** – The member VLANs are additional VLANs that share ports with the master VLAN. The Layer 2 protocol settings for the ports in the master VLAN apply to the same ports in the member VLANs. A change to the master VLAN's Layer 2 protocol configuration or Layer 2 topology affects all the member

VLANs. Member VLANs do not independently run a Layer 2 protocol.

- ***Member VLAN groups*** – A VLAN group is a named set of VLANs. The VLANs within a VLAN group have the same ports and use the same values for other VLAN parameters.

When a Layer 2 topology change occurs on a port in the master VLAN, the same change is applied to that port in all the member VLANs that contain the port. For example, if you configure a topology group whose master VLAN contains ports 1/1 and 1/2, a Layer 2 state change on port 1/1 applies to port 1/1 in all the member VLANs that contain that port. However, the state change does not affect port 1/1 in VLANs that are not members of the topology group.

Control Ports and Free Ports

A port that is in a topology group can be a control port or a free port.

- ***Control port*** – A control port is a port in the master VLAN, and is therefore controlled by the Layer 2 protocol configured in the master VLAN. The same port in all the member VLANs is controlled by the master VLAN's Layer 2 protocol. Each member VLAN must contain all of the control ports and can contain additional ports.
- ***Free port*** – A free port is not controlled by the master VLAN's Layer 2 protocol. The master VLAN can contain free ports. (In this case, the Layer 2 protocol is disabled on those ports.) In addition, any ports in the member VLANs that are not also in the master VLAN are free ports.

NOTE: Since free ports are not controlled by the master port's Layer 2 protocol, they are assumed to always be in the Forwarding state.

Configuration Considerations

- Topology groups are supported in all FESX, FSX, and FWSX devices and associated software releases.
- You must configure the master VLAN and member VLANs or member VLAN groups before you configure the topology group.
- You can configure up to 256 topology groups. Each group can control up to 4096 VLANs. A VLAN cannot be controlled by more than one topology group.
- The topology group must contain a master VLAN and can also contain individual member VLANs, VLAN groups, or a combination of individual member VLANs and VLAN groups.
- Once you add a VLAN as a member of a topology group, all the Layer 2 protocol information on the VLAN is deleted.

Configuring a Topology Group

To configure a topology group, enter commands such as the following:

```
FastIron(config)#topology-group 2
FastIron(config-topo-group-2)#master-vlan 2
FastIron(config-topo-group-2)#member-vlan 3
FastIron(config-topo-group-2)#member-vlan 4
FastIron(config-topo-group-2)#member-vlan 5
FastIron(config-topo-group-2)#member-group 2
```

These commands create topology group 2 and add the following:

- Master VLAN 2
- Member VLANs 2, 3, and 4
- Member VLAN group 2

Syntax: [no] topology-group <group-id>

The <group-id> parameter specifies the topology group ID and can be from 1 – 256.

Syntax: [no] master-vlan <vlan-id>

This command adds the master VLAN. The VLAN must already be configured. Make sure all the Layer 2 protocol settings in the VLAN are correct for your configuration before you add the VLAN to the topology group. A topology group can have only one master VLAN.

NOTE: If you remove the master VLAN (by entering **no master-vlan** <vlan-id>), the software selects the next-highest numbered member VLAN as the new master VLAN. For example, if you remove master VLAN 2 from the example above, the CLI converts member VLAN 3 into the new master VLAN. The new master VLAN inherits the Layer 2 protocol settings of the older master VLAN.

NOTE: If you add a new master VLAN to a topology group that already has a master VLAN, the new master VLAN replaces the older master VLAN. All member VLANs and VLAN groups follow the Layer 2 protocol settings of the new master VLAN.

Syntax: [no] member-vlan <vlan-id>

The <vlan-id> parameter specifies a VLAN ID. The VLAN must already be configured.

Syntax: [no] member-group <num>

The <num> specifies a VLAN group ID. The VLAN group must already be configured.

NOTE: Once you add a VLAN or VLAN group as a member of a topology group, all the Layer 2 protocol configuration information for the VLAN or group is deleted. For example, if STP is configured on a VLAN and you add the VLAN to a topology group, the STP configuration is removed from the VLAN. Once you add the VLAN to a topology group, the VLAN uses the Layer 2 protocol settings of the master VLAN.

If you remove a member VLAN or VLAN group from a topology group, you will need to reconfigure the Layer 2 protocol information in the VLAN or VLAN group.

Displaying Topology Group Information

The following sections show how to display STP information and topology group information for VLANS.

Displaying STP Information

To display STP information for a VLAN, enter a command such as the following:

```
FastIron#show span vlan 4
VLAN 4 BPDU cam_index is 14344 and the Master DMA Are(HEX) 18 1A
STP instance owned by VLAN 2
```

This example shows STP information for VLAN 4. The line shown in bold type indicates that the VLAN’s STP configuration is controlled by VLAN 2. This information indicates that VLAN 4 is a member of a topology group and VLAN 2 is the master VLAN in that topology group.

Displaying Topology Group Information

To display topology group information, enter the following command:

```
FastIron#show topology-group

Topology Group 3
=================
 master-vlan 2
 member-vlan none

 Common control ports          L2 protocol
 ethernet 1/1                  MRP
 ethernet 1/2                  MRP
 ethernet 1/5                  VSRP
 ethernet 2/22                 VSRP
 Per vlan free ports
 ethernet 2/3                   Vlan 2
 ethernet 2/4                   Vlan 2
 ethernet 2/11                  Vlan 2
 ethernet 2/12                  Vlan 2
```

Syntax: show topology-group [<group-id>]

This display shows the following information.

Table 10.1: CLI Display of Topology Group Information

This Field...	Displays...
master-vlan	The master VLAN for the topology group. The settings for STP, MRP, or VSRP on the control ports in the master VLAN apply to all control ports in the member VLANs within the topology group.
member-vlan	The member VLANs in the topology group.
Common control ports	The master VLAN ports that are configured with Layer 2 protocol information. The Layer 2 protocol configuration and state of these ports in the master VLAN applies to the same port numbers in all the member VLANs.
L2 protocol	The Layer 2 protocol configured on the control ports. The Layer 2 protocol can be one of the following: • MRP • STP • VSRP
Per vlan free ports	The ports that are not controlled by the Layer 2 protocol information in the master VLAN.

Metro Ring Protocol (MRP)

Foundry's Metro Ring Protocol (MRP) was introduced in two phases:

- ***MRP Phase 1*** is supported in all software releases and on all Foundry FastIron X Series and FGS devices. See "MRP Rings without Shared Interfaces (MRP Phase 1)" on page 10-6.
- ***MRP Phase 2*** is available in software releases 03.0.00 and later and is supported on all Foundry FastIron X

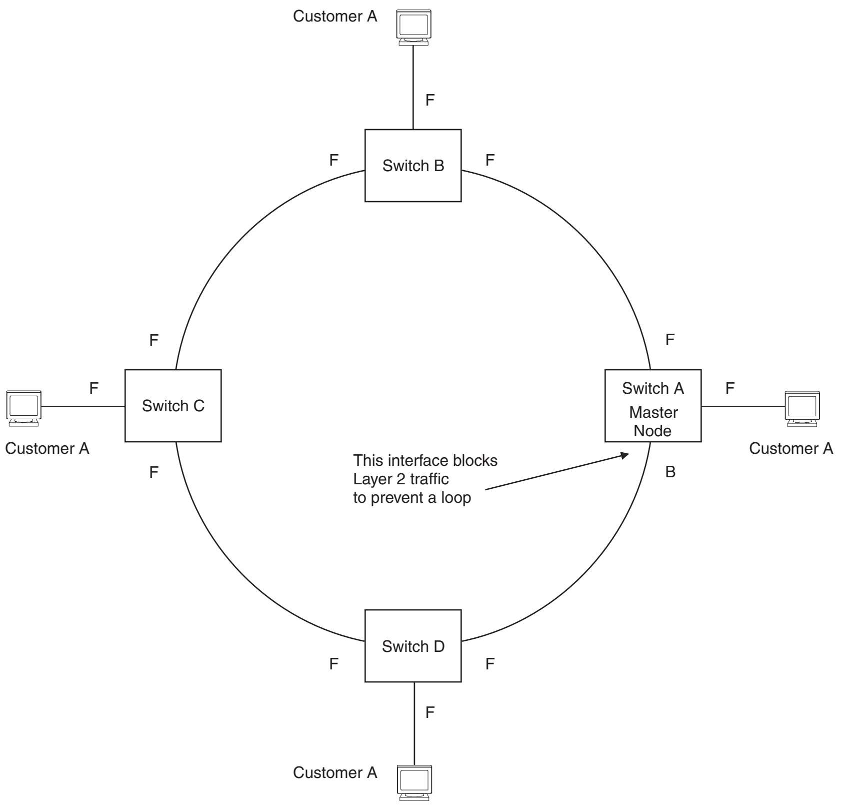
Series devices, except the FESX624 and FESX648. See “MRP Rings with Shared Interfaces (MRP Phase 2)” on page 10-6.

MRP is a Foundry proprietary protocol that prevents Layer 2 loops and provides fast reconvergence in Layer 2 ring topologies. It is an alternative to STP and is especially useful in Metropolitan Area Networks (MANs) where using STP has the following drawbacks:

- STP allows a maximum of seven nodes. Metro rings can easily contain more nodes than this.
- STP has a slow reconvergence time, taking many seconds or even minutes. MRP can detect and heal a break in the ring in sub-second time.

Figure 10.1 shows an example of an MRP metro ring.

Figure 10.1 Metro Ring – Normal State



The ring in this example consists of four MRP nodes (Foundry switches). Each node has two interfaces with the ring. Each node also is connected to a separate customer network. The nodes forward Layer 2 traffic to and from the customer networks through the ring. The ring interfaces are all in one port-based VLAN. Each customer interface can be in the same VLAN as the ring or in a separate VLAN.

One node is configured as the master node of the MRP ring. One of the two interfaces on the master node is configured as the primary interface; the other is the secondary interface. The primary interface originates Ring Health Packets (RHPs), which are used to monitor the health of the ring. An RHP is forwarded on the ring to the next interface until it reaches the secondary interface of the master node. The secondary interface blocks the packet to prevent a Layer 2 loops.

Configuration Notes

- When you configure MRP, Foundry recommends that you disable one of the ring interfaces before beginning the ring configuration. Disabling an interface prevents a Layer 2 loop from occurring while you are configuring MRP on the ring nodes. Once MRP is configured and enabled on all the nodes, you can re-enable the interface.

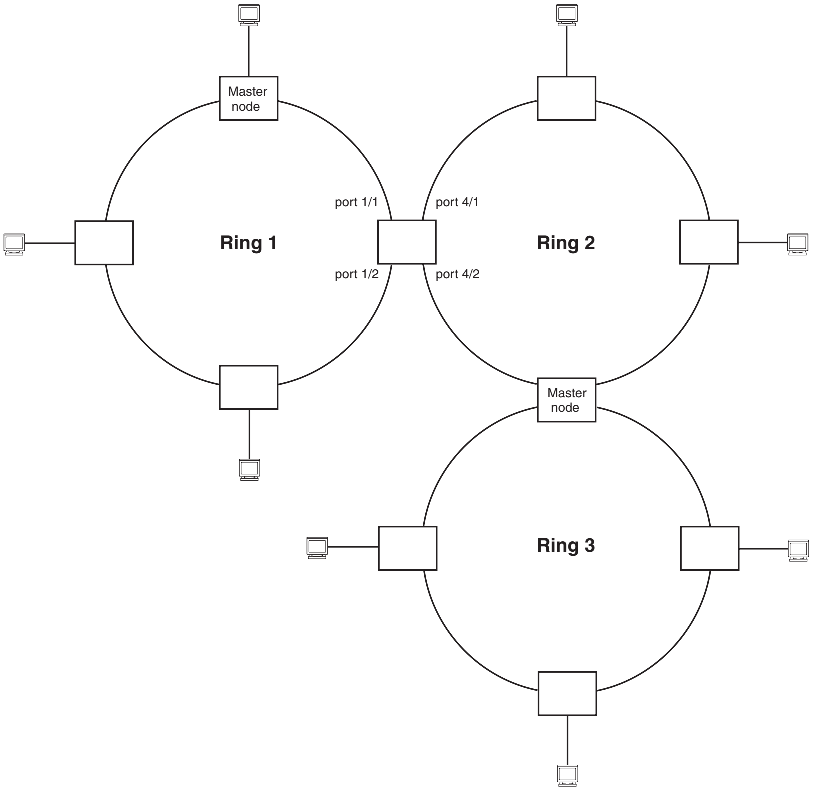
- MRP 2 support was added in release 03.0.00 for the FESX, FSX, FSX 800, FSX 1600, and FWSX.
- MRP 2 is not supported in the FGS, FESX624, and FESX648. These devices support MRP 1 only.
- The above configurations can be configured as MRP masters or MRP members (for different rings).

MRP Rings without Shared Interfaces (MRP Phase 1)

MRP Phase 1 allows you to configure multiple MRP rings, as shown in Figure 10.2, but the rings cannot share the same link. For example, you cannot configure ring 1 and ring 2 to each have interfaces 1/1 and 1/2.

Also, when you configure an MRP ring, any node on the ring can be designated as the master node for the ring. A master node can be the master node of more than one ring. (See Figure 10.2.) Each ring is an independent ring and RHP packets are processed within each ring.

Figure 10.2 Metro Ring – Multiple Rings



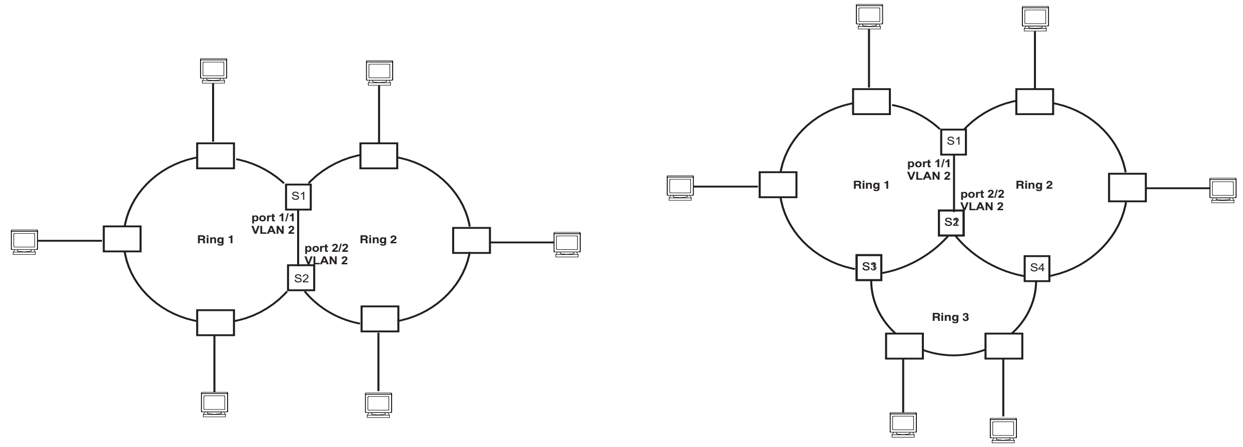
In this example, two nodes are each configured with two MRP rings. Any node in a ring can be the master for its ring. A node also can be the master for more than one ring.

MRP Rings with Shared Interfaces (MRP Phase 2)

NOTE: This feature is supported on FastIron X Series devices running software release 03.0.00 or later. It is supported in the Layer 3 router code and in the Layer 2 switch code.

With MRP Phase 2, MRP rings can be configured to share the same interfaces as long as the interfaces belong to the same VLAN. Figure 10.3 shows examples of multiple MRP rings that share the same interface.

Figure 10.3 Examples of Multiple Rings Sharing the Same Interface - MRP Phase 2



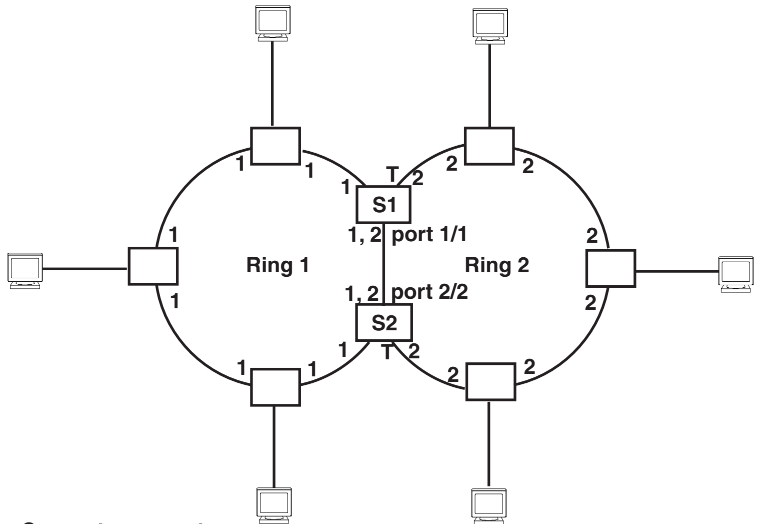
Example 1

Example 2

On each node that will participate in the ring, you specify the ring's ID and the interfaces that will be used for ring traffic. In a multiple ring configuration, a ring's ID determines its priority. The lower the ring ID, the higher priority of a ring.

A ring's ID is also used to identify the interfaces that belong to a ring.

Figure 10.4 Interface IDs and Types



C = customer port

For example, in Figure 10.4, the ID of all interfaces on all nodes on Ring 1 is 1 and all interfaces on all nodes on Ring 2 is 2. Port 1/1 on node S1 and Port 2/2 on S2 have the IDs of 1 and 2 since the interfaces are shared by Rings 1 and 2.

The ring's ID is also used to determine an interface's priority. Generally, a ring's ID is also the ring's priority and the priority of all interfaces on that ring. However, if the interface is shared by two or more rings, then the highest priority (lowest ID) becomes the priority of the interface. For example, in Figure 10.4, all interfaces on Ring 1, except for Port 1/1 on node S1 and Port 2/2 on node S2 have a priority of 1. Likewise, all interfaces on Ring 2, except for Port 1/1 on node S1 and Port 2/2 on node S2 have a priority of 2. Port 1/1 on S1 and Port 2/2 on S2 have a priority of 1 since 1 is the highest priority (lowest ID) of the rings that share the interface.

If a node has interfaces that have different IDs, the interfaces that belong to the ring with the highest priority become regular ports. Those interfaces that do not belong to the ring with the highest priority become tunnel ports. In Figure 10.4, nodes S1 and S2 have interfaces that belong to Rings 1 and 2. Those interfaces with a priority of 1 are regular ports. The interfaces with a priority of 2 are the tunnel ports since they belong to Ring 2, which has a lower priority than Ring 1.

Selection of Master Node

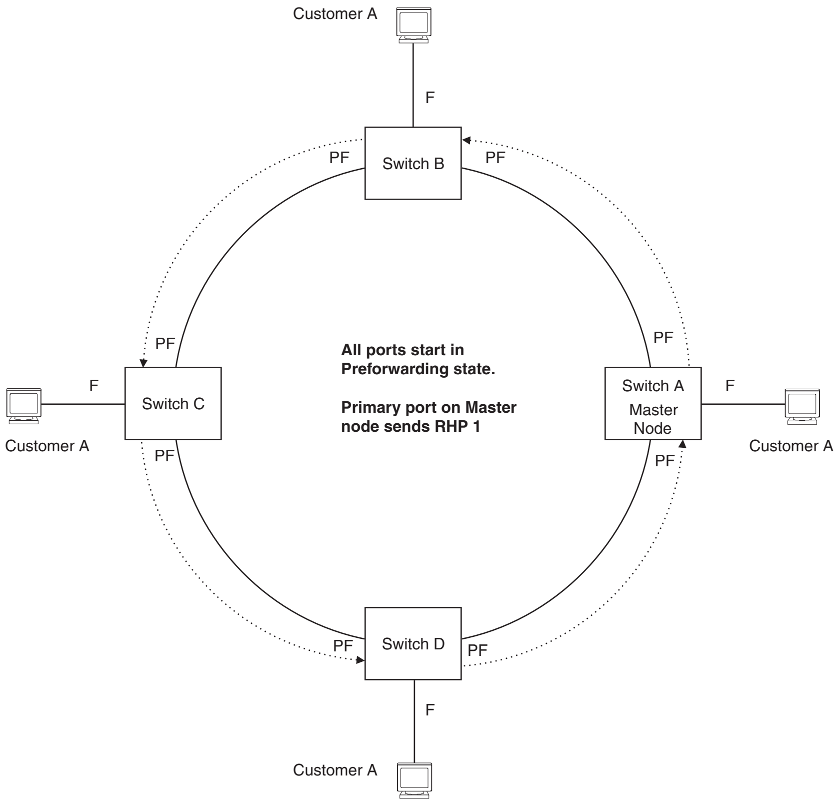
Allowing MRP rings to share interfaces limits the nodes that can be designated as the master node. Any node on an MRP ring that does not have a shared interface can be designated as the ring's master node. However, if all nodes on the ring have shared interfaces, nodes that do not have tunnel ports can be designated as the master node of that ring. If none of the nodes meet these criteria, you must change the rings' priorities by reconfiguring the rings' ID.

In Figure 10.4, any of the nodes on Ring 1, even S1 or S2, can be a master node since none of its interfaces are tunnel ports. However in Ring 2, neither S1 nor S2 can be a master node since these nodes contain tunnel ports.

Ring Initialization

The ring shown in Figure 10.1 shows the port states in a fully initialized ring without any broken links. Figure 10.5 shows the initial state of the ring, when MRP is first enabled on the ring's switches. All ring interfaces on the master node and member nodes begin in the Preforwarding state (PF).

Figure 10.5 Metro Ring – Initial State



MRP uses Ring Health Packets (RHPs) to monitor the health of the ring. An RHP is an MRP protocol packet. The source address is the MAC address of the master node and the destination MAC address is a protocol address for MRP. The Master node generates RHPs and sends them on the ring. The state of a ring port depends on the RHPs.

RHP Processing in MRP Phase 1

A ring interface can have one of the following MRP states:

- Preforwarding (PF) – The interface can forward RHPS but cannot forward data. All ring ports begin in this state when you enable MRP.

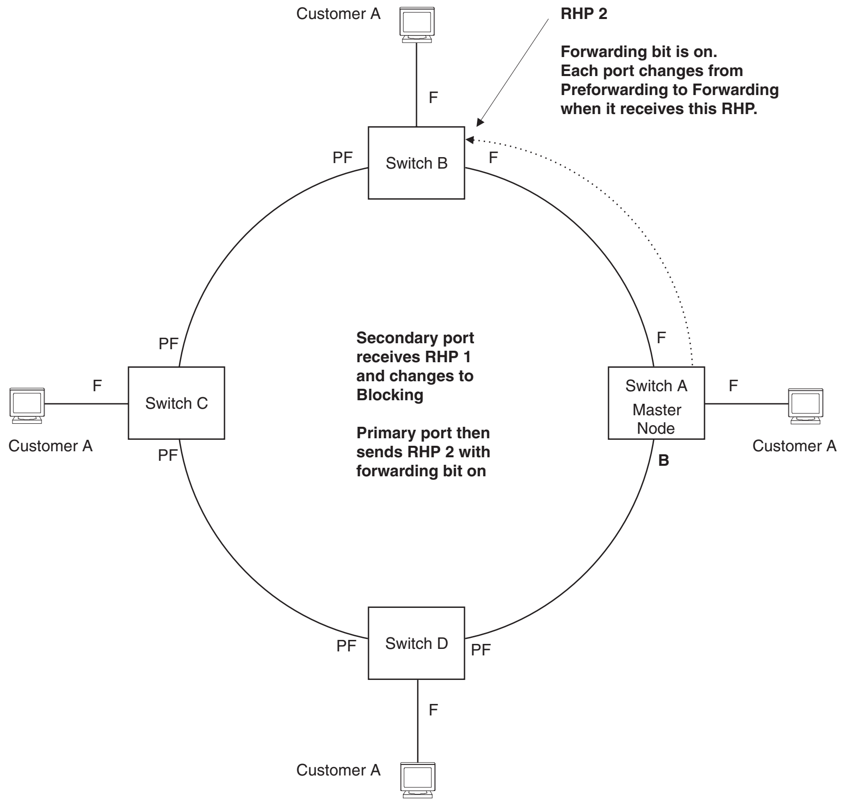
- Forwarding (F) – The interface can forward data as well as RHPs. An interface changes from Preforwarding to Forwarding when the port's preforwarding time expires. This occurs if the port does not receive an RHP from the Master, or if the forwarding bit in the RHPs received by the port is off. This indicates a break in the ring. The port heals the ring by changing its state to Forwarding. The preforwarding time is the number of milliseconds the port will remain in the Preforwarding state before changing to the Forwarding state, even without receiving an RHP.
- Blocking (B) – The interface cannot forward data. Only the secondary interface on the Master node can be Blocking.

When MRP is enabled, all ports begin in the Preforwarding state. The primary interface on the Master node, although it is in the Preforwarding state like the other ports, immediately sends an RHP onto the ring. The secondary port on the Master node listens for the RHP.

- If the secondary port receives the RHP, all links in the ring are up and the port changes its state to Blocking. The primary port then sends another MRP with its forwarding bit set on. As each of the member ports receives the RHP, the ports changes their state to Forwarding. Typically, this occurs in sub-second time. The ring very quickly enters the fully initialized state.
- If the secondary port does not receive the RHP by the time the preforwarding time expires, a break has occurred in the ring. The port changes its state to Forwarding. The member ports also change their states from Preforwarding to Forwarding as their preforwarding timers expire. The ring is not intact, but data can still travel among the nodes using the links that are up.

Figure 10.6 shows an example.

Figure 10.6 Metro Ring – from Preforwarding to Forwarding

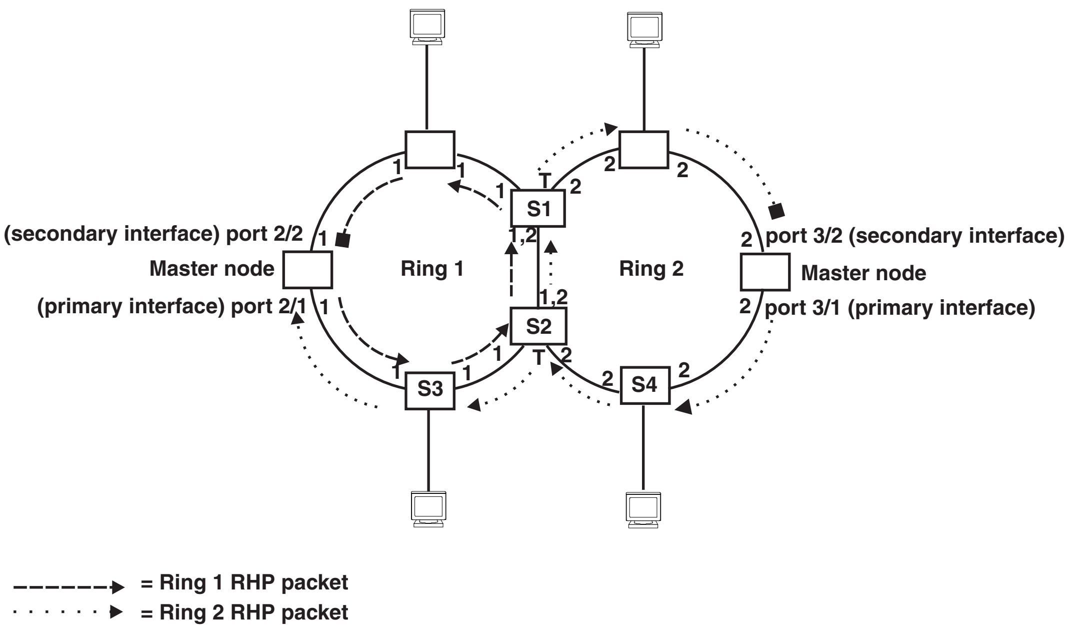


Each RHP also has a sequence number. MRP can use the sequence number to determine the round-trip time for RHPs in the ring. See "Using MRP Diagnostics" on page 10-15.

RHP Processing in MRP Phase 2

Figure 10.7 shows an example of how RHP packets are processed normally in MRP rings with shared interfaces.

Figure 10.7 Flow of RHP Packets on MRP Rings with Shared Interfaces


T
2
2
1
1
1
S1
2
(secondary interface) port 2/2
1
1,2
2
port 3/2 (secondary interface)
Master node
Ring 1
Ring 2
Master node
(primary interface) port 2/1
1
1,2
2
port 3/1 (primary interface)
S2
1
T
2
1
1
2
S4
2
S3
= Ring 1 RHP packet
= Ring 2 RHP packet

Port 2/1 on Ring 1's master node is the primary interface of the master node. The primary interface forwards an RHP packet on the ring. Since all the interfaces on Ring 1 are regular ports, the RHP packet is forwarded to all the interfaces until it reaches Port 2/2, the secondary interface of the master node. Port 2/2 then blocks the packet to complete the process.

On Ring 2, Port 3/1, is the primary interface of the master node. It sends an RHP packet on the ring. Since all ports on S4 are regular ports, the RHP packet is forwarded on those interfaces. When the packet reaches S2, the receiving interface is a tunnel port. The port compares the packet's priority to its priority. Since the packet's priority is the same as the tunnel port's priority, the packet is forwarded up the link shared by Rings 1 and 2.

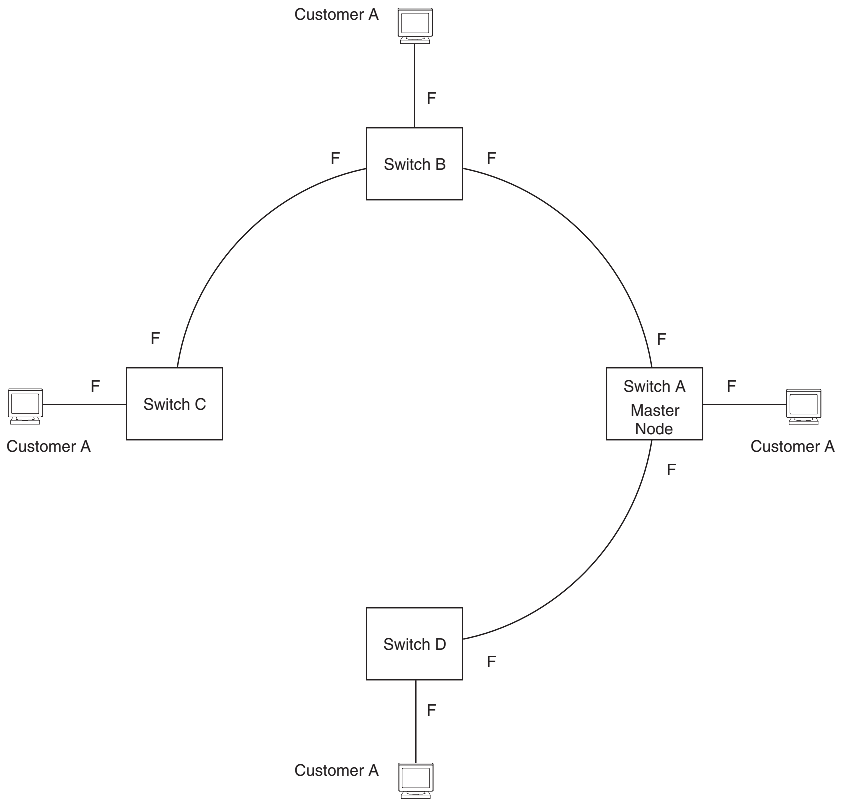
When the RHP packet reaches the interface on node S2 shared by Rings 1 and 2, the packet is forwarded since its priority is less than the interface's priority. The packet continues to be forwarded to node S1 until it reaches the tunnel port on S1. That tunnel port determines that the RHP packet's priority is equal to the port's priority and forwards the packet. The RHP packet is forwarded to the remaining interfaces on Ring 2 until it reaches port 3/2, the secondary interface of the master node. Port 3/2 then blocks the packet to prevent a loop.

When the RHP packet from Ring 2 reached S2, it was also forwarded from S2 to S3 on Ring 1 since the port on S2 has a higher priority than the RHP packet. The packets is forwarded around Ring 1 until it reaches port 2/2, Ring 1's the secondary port. The RHP packet is then blocked by that port.

How Ring Breaks Are Detected and Healed

Figure 10.8 shows ring interface states following a link break. MRP quickly heals the ring and preserves connectivity among the customer networks.

Figure 10.8 Metro Ring – Ring Break



If a break in the ring occurs, MRP heals the ring by changing the states of some of the ring interfaces.

- Blocking interface – The Blocking interface on the Master node has a dead timer. If the dead time expires before the interface receives one of its ring's RHPs, the interface changes state to Preforwarding. Once the secondary interface changes state to Preforwarding:
  - If the interface receives an RHP, the interface changes back to the Blocking state and resets the dead timer.
  - If the interface does not receive an RHP for its ring before the Preforwarding time expires, the interface changes to the Forwarding state, as shown in Figure 10.8.
- Forwarding interfaces – Each member interface remains in the Forwarding state.

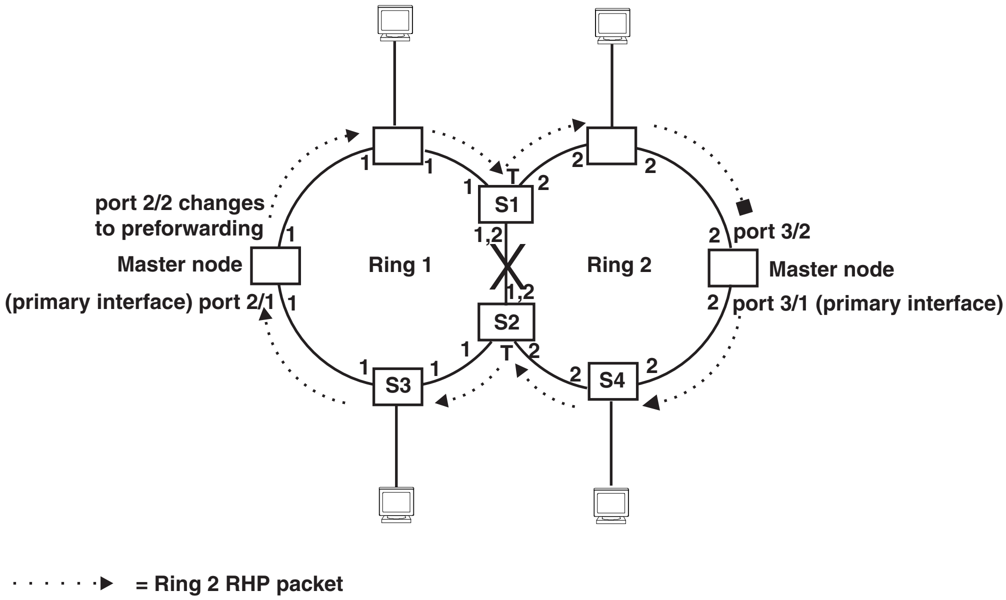
When the broken link is repaired, the link's interfaces come up in the Preforwarding state, which allows RHPs to travel through the restored interfaces and reach the secondary interface on the Master node.

- If an RHP reaches the Master node's secondary interface, the ring is intact. The secondary interface changes to Blocking. The Master node sets the forwarding bit on in the next RHP. When the restored interfaces receive this RHP, they immediately change state to Forwarding.
- If an RHP does not reach the Master node's secondary interface, the ring is still broken. The Master node does not send an RHP with the forwarding bit on. In this case, the restored interfaces remain in the Preforwarding state until the preforwarding timer expires, then change to the Forwarding state.

If the link between ***shared interfaces*** breaks (Figure 10.9), the secondary interface on Ring 1's master node changes to a preforwarding state. The RHP packet sent by port 3/1 on Ring 2 is forwarded through the interfaces on S4, then to S2. The packet is then forwarded through S2 to S3, but not from S2 to S1 since the link between the two nodes is not available. When the packet reaches Ring 1's master node, the packet is forwarded through the secondary interface since it is currently in a preforwarding state. A secondary interface in preforwarding mode ignores any RHP packet that is not from its ring. The secondary interface changes to blocking mode only when the RHP packet forwarded by its primary interface is returned.

The packet then continues around Ring 1, through the interfaces on S1 to Ring 2 until it reaches Ring 2's master node. Port 3/2, the secondary interface on Ring 2 changes to blocking mode since it received its own packet, then blocks the packet to prevent a loop.

Figure 10.9 Flow of RHP Packets When a Link for Shared Interfaces Breaks

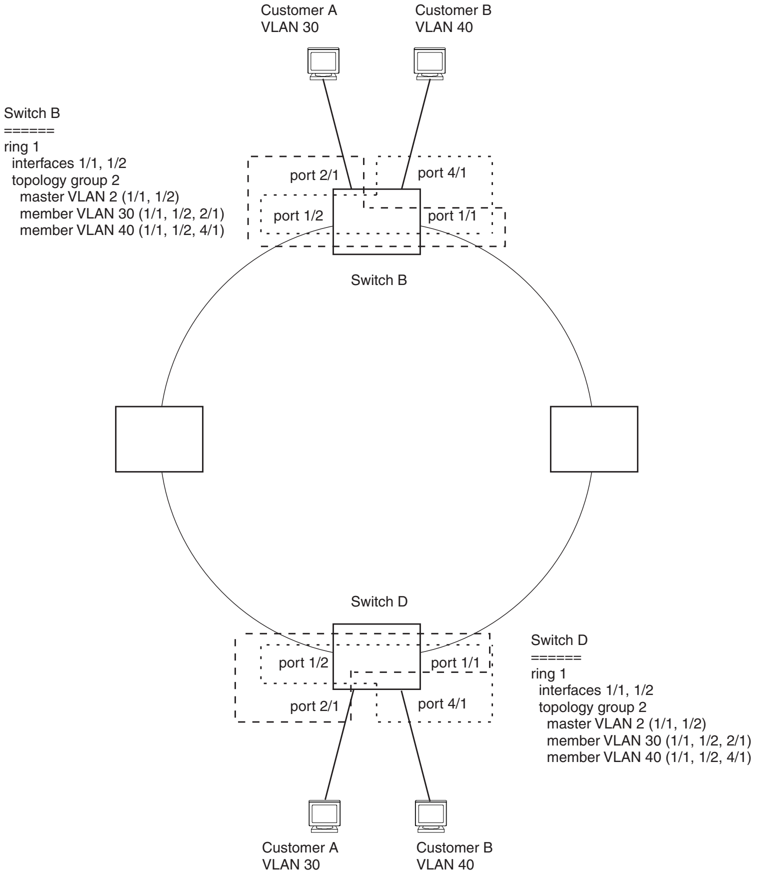


RHP packets follow this flow until the link is restored; then the RHP packet returns to it normal flow as shown in Figure 10.7.

Master VLANs and Customer VLANs

All the ring ports must be in the same VLAN. Placing the ring ports in the same VLAN provides Layer 2 connectivity for a given customer across the ring. Figure 10.10 shows an example.

Figure 10.10 Metro Ring – Ring VLAN and Customer VLANs



Notice that each customer has their own VLAN. Customer A has VLAN 30 and Customer B has VLAN 40. Customer A's host attached to Switch D can reach the Customer A host attached to Switch B at Layer 2 through the ring. Since Customer A and Customer B are on different VLANs, they will not receive each other's traffic.

You can configure MRP separately on each customer VLAN. However, this is impractical if you have many customers. To simplify configuration when you have a lot of customers (and therefore a lot of VLANs), you can use a topology group.

A topology group enables you to control forwarding in multiple VLANs using a single instance of a Layer 2 protocol such as MRP. A topology group contains a master VLAN and member VLANs. The master VLAN contains all the configuration parameters for the Layer 2 protocol (STP, MRP, or VSRP). The member VLANs use the Layer 2 configuration of the master VLAN.

In Figure 10.10, VLAN 2 is the master VLAN and contains the MRP configuration parameters for ring 1. VLAN 30 and VLAN 40, the customer VLANs, are member VLANs in the topology group. Since a topology group is used, a single instance of MRP provides redundancy and loop prevention for both the customer VLANs.

If you use a topology group:

- The master VLAN must contain the ring interfaces. The ports must be tagged, since they will be shared by multiple VLANs.
- The member VLAN for a customer must contain the two ring interfaces and the interfaces for the customer. Since these interfaces are shared with the master VLAN, they must be tagged. Do not add another customer's interfaces to the VLAN.

For more information about topology groups, see "Topology Groups" on page 10-1.

See "MRP CLI Example" on page 10-19 for the configuration commands required to implement the MRP configuration shown in Figure 10.10.

Configuring MRP

To configure MRP, perform the following tasks. You need to perform the first task on only one of the nodes. Perform the remaining tasks on all the nodes.

NOTE: There are no new commands or parameters to configure MRP with shared interfaces (MRP Phase 2).

- Disable one of the ring interfaces. This prevents a Layer 2 loop from occurring while you are configuring the devices for MRP.
- Add an MRP ring to a port-based VLAN. When you add a ring, the CLI changes to the configuration level for the ring, where you can perform the following tasks.
  - Optionally, specify a name for the ring.
  - On the master node only, enable the device to be the master for the ring. Each ring can have only one master node.
  - Specify the MRP interfaces. Each device has two interfaces to an MRP ring.
  - Optionally, change the hello time and the preforwarding time. These parameters control how quickly failover occurs following a change in the state of a link in the ring.
  - Enable the ring.
- Optionally, add the ring's VLAN to a topology group to add more VLANs to the ring. If you use a topology group, make sure you configure MRP on the group's master VLAN. See "Topology Groups" on page 10-1.
- Re-enable the interface you disabled to prevent a Layer 2 loop. Once MRP is enabled, MRP will prevent the Layer 2 loop.

Adding an MRP Ring to a VLAN

To add an MRP ring to a VLAN, enter commands such as the following.

NOTE: If you plan to use a topology group to add VLANs to the ring, make sure you configure MRP on the topology group's master VLAN.

```
FastIron(config)#vlan 2
FastIron(config-vlan-2)#metro-ring 1
FastIron(config-vlan-2-mrp-1)#name CustomerA
FastIron(config-vlan-2-mrp-1)#master
FastIron(config-vlan-2-mrp-1)#ring-interface ethernet 1/1 ethernet 1/2
FastIron(config-vlan-2-mrp-1)#enable
```

These commands configure an MRP ring on VLAN 2. The ring ID is 1, the ring name is CustomerA, and this node (this Foundry device) is the master for the ring. The ring interfaces are 1/1 and 1/2. Interface 1/1 is the primary interface and 1/2 is the secondary interface. The primary interface will initiate RHPs by default. The ring takes effect in VLAN 2.

To configure MRP rings with shared interfaces, enter commands such as the following:

```
FastIron(config)#vlan 2
FastIron(config-vlan-2)#metro-ring 1
FastIron(config-vlan-2-mrp-1)#name CustomerA
FastIron(config-vlan-2-mrp-1)#ring-interface ethernet 1/1 ethernet 1/2
FastIron(config-vlan-2-mrp-1)#enable
FastIron(config-vlan-2-mrp-1)#metro-ring 2
FastIron(config-vlan-2-mrp-2)#name CustomerB
FastIron(config-vlan-2-mrp-2)#ring-interface ethernet 1/1 ethernet 1/2
FastIron(config-vlan-2-mrp-2)#enable
```

Syntax: [no] metro-ring <ring-id>

The <ring-id> parameter specifies the ring ID. In software releases prior to 03.0.00 and in the FGS, the ring ID can be a value from 1 – 255. Starting in release 03.0.00, the <ring-id> can be from 1 – 1023; ID 256 is reserved for VSRP.

Syntax: [no] name <string>

The <string> parameter specifies a name for the ring. The name is optional, but it can be up to 20 characters long and can include blank spaces. If you use a name that has blank spaces, enclose the name in double quotation marks (for example: “Customer A”).

Syntax: [no] master

Configures this node as the master node for the ring. Enter this command only on one node in the ring. The node is a member (non-master) node by default.

Syntax: [no] ring-interface ethernet <primary-if> ethernet <secondary-if>

The **ethernet** <primary-if> parameter specifies the primary interface. On the master node, the primary interface is the one that originates RHPs. Ring control traffic and Layer 2 data traffic will flow in the outward direction from this interface by default. On member nodes, the direction of traffic flow depends on the traffic direction selected by the master node. Therefore, on a member node, the order in which you enter the interfaces does not matter.

The **ethernet** <secondary-if> parameter specifies the secondary interface.

NOTE: To take advantage of every interface in a Metro network, you can configure another MRP ring and either configure a different Master node for the ring or reverse the configuration of the primary and secondary interfaces on the Master node. Configuring multiple rings enables you to use all the ports in the ring. The same port can forward traffic one ring while blocking traffic for another ring.

Syntax: [no] enable

The **enable** command enables the ring.

Changing the Hello and PreForwarding Times

You also can change the RHP hello time and preforwarding time. To do so, enter commands such as the following:

```
FastIron(config-vlan-2-mrp-1)#hello-time 200
FastIron(config-vlan-2-mrp-1)#preforwarding-time 400
```

These commands change the hello time to 200 ms and change the preforwarding time to 400 ms.

Syntax: [no] hello-time <ms>

Syntax: [no] preforwarding-time <ms>

The <ms> specifies the number of milliseconds. For the hello time, you can specify from 100 – 1000 (one second). The default hello time is 100 ms. The preforwarding time can be from 200 – 5000 ms, but must be at least twice the value of the hello time and must be a multiple of the hello time. The default preforwarding time is 300 ms. A change to the hello time or preforwarding time takes effect as soon as you enter the command.

Configuration Notes

- The preforwarding time must be at least twice the value of the hello time and must be a multiple of the hello time.
- If UDLD is also enabled on the device, Foundry recommends that you set the MRP preforwarding time slightly higher than the default of 300 ms; for example, to 400 or 500 ms.
- You can use MRP ring diagnostics to determine whether you need to change the hello time and preforwarding time. See “Using MRP Diagnostics” .

Using MRP Diagnostics

The MRP diagnostics feature calculates how long it takes for RHP packets to travel through the ring. When you enable MRP diagnostics, the software tracks RHP packets according to their sequence numbers and calculates how long it takes an RHP packet to travel one time through the entire ring. When you display the diagnostics, the

CLI shows the average round-trip time for the RHP packets sent since you enabled diagnostics. The calculated results have a granularity of 1 microsecond.

Enabling MRP Diagnostics

To enable MRP diagnostics for a ring, enter the following command on the Master node, at the configuration level for the ring:

```
FastIron(config-vlan-2-mrp-1)#diagnostics
```

Syntax: [no] diagnostics

NOTE: This command is valid only on the master node.

Displaying MRP Diagnostics

To display MRP diagnostics results, enter the following command on the Master node:

```
FastIron#show metro 1 diag

Metro Ring 1 - CustomerA
=============
diagnostics results

Ring      Diag      RHP average     Recommended     Recommended
id        state     time(microsec)  hello time(ms)  Prefwing time(ms)
2         enabled   125             100             300

Diag frame sent     Diag frame lost
1230                0
```

Syntax: show metro <ring-id> diag

This display shows the following information.

Table 10.2:CLI Display of MRP Ring Diagnostic Information

This Field...	Displays...
Ring id	The ring ID.
Diag state	The state of ring diagnostics.
RHP average time	The average round-trip time for an RHP packet on the ring. The calculated time has a granularity of 1 microsecond.
Recommended hello time	The hello time recommended by the software based on the RHP average round-trip time.
Recommended Prefwing time	The preforwarding time recommended by the software based on the RHP average round-trip time.
Diag frame sent	The number of diagnostic RHPs sent for the test.
Diag frame lost	The number of diagnostic RHPs lost during the test.

If the recommended hello time and preforwarding time are different from the actual settings and you want to change them, see “Configuring MRP” on page 10-14.

Displaying MRP Information

You can display the following MRP information:

- Topology group configuration information
- Ring configuration information and statistics

Displaying Topology Group Information

To display topology group information, enter the following command:

Syntax: show topology-group [<group-id>]

See "Displaying Topology Group Information" on page 10-3 for more information.

Displaying Ring Information

To display ring information, enter the following command:

```
FastIron#show metro

Metro Ring 1
=============
Ring       State     Ring       Master     Topo       Hello      Prefwing
id                   role       vlan       group      time(ms)   time(ms)
2          enabled   member     2          not conf   100        300

Ring interfaces      Interface role      Forwarding state    Active interface    Interface Type
ethernet 1/1         primary             disabled            none                Regular
ethernet 1/2         secondary           forwarding          ethernet 2          Tunnel

RHPs sent            RHPs rcvd           TC RHPs rcvd        State changes
3                      0                     0                      4
```

Syntax: show metro [<ring-id>]

This display shows the following information.

Table 10.3: CLI Display of MRP Ring Information

This Field...	Displays...
Ring id	The ring ID
State	The state of MRP. The state can be one of the following: • enabled – MRP is enabled • disabled – MRP is disabled
Ring role	Whether this node is the master for the ring. The role can be one of the following: • master • member
Master vlan	The ID of the master VLAN in the topology group used by this ring. If a topology group is used by MRP, the master VLAN controls the MRP settings for all VLANs in the topology group. **Note**: The topology group ID is 0 if the MRP VLAN is not the master VLAN in a topology group. Using a topology group for MRP configuration is optional.

Table 10.3: CLI Display of MRP Ring Information (Continued)

This Field...	Displays...
Topo group	The topology group ID.
Hello time	The interval, in milliseconds, at which the Forwarding port on the ring's master node sends Ring Hello Packets (RHPs).
Prefwing time	The number of milliseconds an MRP interface that has entered the Preforwarding state will wait before changing to the Forwarding state. If a member port in the Preforwarding state does not receive an RHP within the Preforwarding time (Prefwing time), the port assumes that a topology change has occurred and changes to the Forwarding state. The secondary port on the Master node changes to Blocking if it receives an RHP, but changes to Forwarding if the port does not receive an RHP before the preforwarding time expires. **Note**: A member node's Preforwarding interface also changes from Preforwarding to Forwarding if it receives an RHP whose forwarding bit is on.
Ring interfaces	The device's two interfaces with the ring. **Note**: If the interfaces are trunk groups, only the primary ports of the groups are listed.
Interface role	The interface role can be one of the following: • primary • Master node – The interface generates RHPs. • Member node – The interface forwards RHPs received on the other interface (the secondary interface). • secondary – The interface does not generate RHPs. • Master node – The interface listens for RHPs. • Member node – The interface receives RHPs.
Forwarding state	Whether MRP Forwarding is enabled on the interface. The forwarding state can be one of the following: • blocking – The interface is blocking Layer 2 data traffic and RHPs • disabled – The interface is down • forwarding – The interface is forwarding Layer 2 data traffic and RHPs • preforwarding – The interface is listening for RHPs but is blocking Layer 2 data traffic
Active interface	The physical interfaces that are sending and receiving RHPs. **Note**: If a port is disabled, its state is shown as "disabled". **Note**: If an interface is a trunk group, only the primary port of the group is listed.
Interface Type	Shows if the interface is a regular port or a tunnel port.

Table 10.3: CLI Display of MRP Ring Information (Continued)

This Field...	Displays...
RHPs sent	The number of RHPs sent on the interface. **Note**: This field applies only to the master node. On non-master nodes, this field contains 0. This is because the RHPs are forwarded in hardware on the non-master nodes.
RHPs rcvd	The number of RHPs received on the interface. **Note**: On most Foundry devices, this field applies only to the master node. On non-master nodes, this field contains 0. This is because the RHPs are forwarded in hardware on the non-master nodes. However, on the FastIron devices, the RHP received counter on non-master MRP nodes increment. This is because, on FastIron devices, the CPU receives a copy of the RHPs forwarded in hardware.
TC RHPs rcvd	The number of Topology Change RHPs received on the interface. A Topology Change RHP indicates that the ring topology has changed.
State changes	The number of MRP interface state changes that have occurred. The state can be one of the states listed in the Forwarding state field.
Interface Type	Shows if the interface is a regular port or a tunnel port.

MRP CLI Example

The following examples show the CLI commands required to implement the MRP configuration shown in Figure 10.10 on page 10-13.

NOTE: For simplicity, the figure shows the VLANs on only two switches. The CLI examples implement the ring on all four switches.

Commands on Switch A (Master Node)

The following commands configure a VLAN for the ring. The ring VLAN must contain both of the node’s interfaces with the ring. Add these interfaces as tagged interfaces, since the interfaces also must be in each of the customer VLANs configured on the node.

```
FastIron(config)#vlan 2
FastIron(config-vlan-2)#tag ethernet 1/1 to 1/2
FastIron(config-vlan-2)#metro-ring 1
FastIron(config-vlan-2-mrp-1)#name “Metro A”
FastIron(config-vlan-2-mrp-1)#master
FastIron(config-vlan-2-mrp-1)#ring-interface ethernet 1/1 ethernet 1/2
FastIron(config-vlan-2-mrp-1)#enable
FastIron(config-vlan-2-mrp-1)#exit
FastIron(config-vlan-2)#exit
```

The following commands configure the customer VLANs. The customer VLANs must contain both the ring interfaces as well as the customer interfaces.

```
FastIron(config)#vlan 30
FastIron(config-vlan-30)#tag ethernet 1/1 to 1/2
FastIron(config-vlan-30)#tag ethernet 2/1
FastIron(config-vlan-30)#exit
FastIron(config)#vlan 40
FastIron(config-vlan-40)#tag ethernet 1/1 to 1/2
FastIron(config-vlan-40)#tag ethernet 4/1
FastIron(config-vlan-40)#exit
```

The following commands configure topology group 1 on VLAN 2. The master VLAN is the one that contains the MRP configuration. The member VLANs use the MRP parameters of the master VLAN. The control interfaces (the ones shared by the master VLAN and member VLAN) also share MRP state.

```
FastIron(config)#topology-group 1
FastIron(config-topo-group-1)#master-vlan 2
FastIron(config-topo-group-1)#member-vlan 30
FastIron(config-topo-group-1)#member-vlan 40
```

Commands on Switch B

The commands for configuring Switches B, C, and D are similar to the commands for configuring Switch A, with two differences: the nodes are not configured to be the ring master. Omitting the **master** command is required for non-master nodes.

```
FastIron(config)#vlan 2
FastIron(config-vlan-2)#tag ethernet 1/1 to 1/2
FastIron(config-vlan-2)#metro-ring 1
FastIron(config-vlan-2-mrp-1)#name “Metro A”
FastIron(config-vlan-2-mrp-1)#ring-interface ethernet 1/1 ethernet 1/2
FastIron(config-vlan-2-mrp-1)#enable
FastIron(config-vlan-2)#exit

FastIron(config)#vlan 30
FastIron(config-vlan-30)#tag ethernet 1/1 to 1/2
FastIron(config-vlan-30)#tag ethernet 2/1
FastIron(config-vlan-30)#exit
FastIron(config)#vlan 40
FastIron(config-vlan-40)#tag ethernet 1/1 to 1/2
FastIron(config-vlan-40)#tag ethernet 4/1
FastIron(config-vlan-40)#exit

FastIron(config)#topology-group 1
FastIron(config-topo-group-1)#master-vlan 2
FastIron(config-topo-group-1)#member-vlan 30
FastIron(config-topo-group-1)#member-vlan 40
```

Commands on Switch C

```
FastIron(config)#vlan 2
FastIron(config-vlan-2)#tag ethernet 1/1 to 1/2
FastIron(config-vlan-2)#metro-ring 1
FastIron(config-vlan-2-mrp-1)#name “Metro A”
FastIron(config-vlan-2-mrp-1)#ring-interface ethernet 1/1 ethernet 1/2
FastIron(config-vlan-2-mrp-1)#enable
FastIron(config-vlan-2)#exit

FastIron(config)#vlan 30
FastIron(config-vlan-30)#tag ethernet 1/1 to 1/2
FastIron(config-vlan-30)#tag ethernet 2/1
FastIron(config-vlan-30)#exit
FastIron(config)#vlan 40
FastIron(config-vlan-40)#tag ethernet 1/1 to 1/2
FastIron(config-vlan-40)#tag ethernet 4/1
FastIron(config-vlan-40)#exit

FastIron(config)#topology-group 1
FastIron(config-topo-group-1)#master-vlan 2
FastIron(config-topo-group-1)#member-vlan 30
FastIron(config-topo-group-1)#member-vlan 40
```

Commands on Switch D

```
FastIron(config)#vlan 2
FastIron(config-vlan-2)#tag ethernet 1/1 to 1/2
FastIron(config-vlan-2)#metro-ring 1
FastIron(config-vlan-2-mrp-1)#name “Metro A”
FastIron(config-vlan-2-mrp-1)#ring-interface ethernet 1/1 ethernet 1/2
FastIron(config-vlan-2-mrp-1)#enable
FastIron(config-vlan-2)#exit

FastIron(config)#vlan 30
FastIron(config-vlan-30)#tag ethernet 1/1 to 1/2
FastIron(config-vlan-30)#tag ethernet 2/1
FastIron(config-vlan-30)#exit
FastIron(config)#vlan 40
FastIron(config-vlan-40)#tag ethernet 1/1 to 1/2
FastIron(config-vlan-40)#tag ethernet 4/1
FastIron(config-vlan-40)#exit

FastIron(config)#topology-group 1
FastIron(config-topo-group-1)#master-vlan 2
FastIron(config-topo-group-1)#member-vlan 30
FastIron(config-topo-group-1)#member-vlan 40
```

Virtual Switch Redundancy Protocol (VSRP)

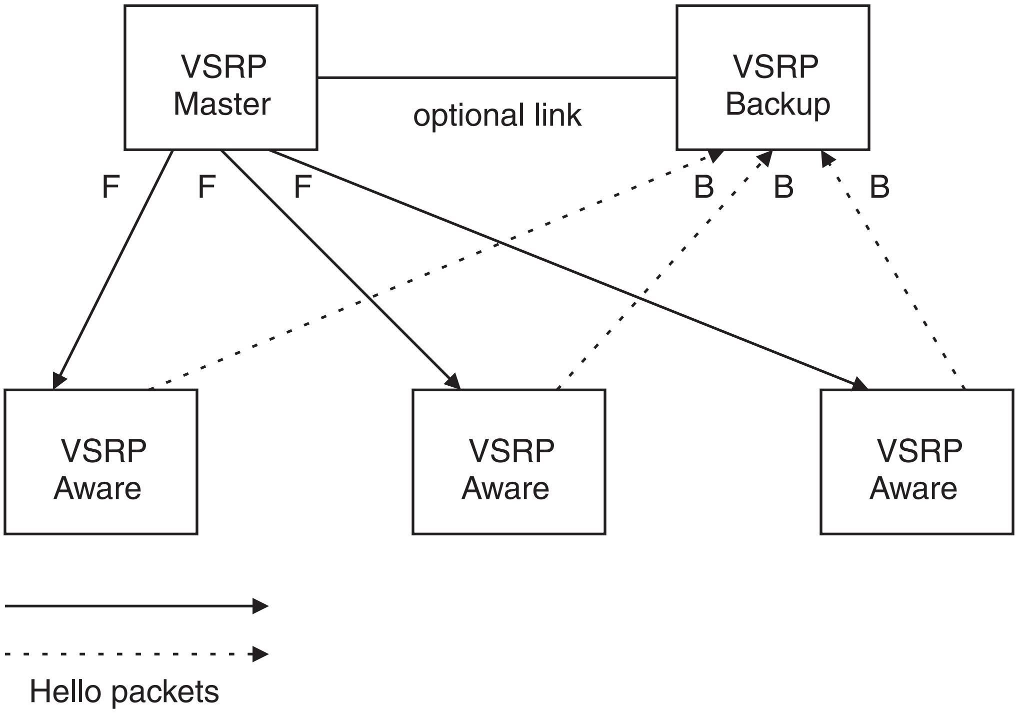
Virtual Switch Redundancy Protocol (VSRP) is a Foundry proprietary protocol that provides redundancy and sub-second failover in Layer 2 and Layer 3 mesh topologies. Based on the Foundry Virtual Router Redundancy Protocol Extended (VRRPE), VSRP provides one or more backups for a Layer 2 Switch or Layer 3 Switch. If the active Layer 2 Switch or Layer 3 Switch becomes unavailable, one of the backups takes over as the active device and continues forwarding traffic for the network.

The FastIron family of switches support full VSRP as well as ***VSRP-awareness***. A Foundry device that is not itself configured for VSRP but is connected to a Foundry device that is configured for VSRP, is ***VSRP aware***.

You can use VSRP for Layer 2, Layer 3, or for both layers. On Layer 3 Switches, Layer 2 and Layer 3 share the same VSRP configuration information. On Layer 2 Switches, VSRP applies only to Layer 2.

Figure 10.11 shows an example of a VSRP configuration.

Figure 10.11 VSRP Mesh – Redundant Paths for Layer 2 and Layer 3 Traffic



In this example, two Foundry devices are configured as redundant paths for VRID 1. On each of the devices, a Virtual Router ID (VRID) is configured on a port-based VLAN. Since VSRP is primarily a Layer 2 redundancy protocol, the VRID applies to the entire VLAN. However, you can selectively remove individual ports from the VRID if needed.

Following Master election (described below), one of the Foundry devices becomes the Master for the VRID and sets the state of all the VLAN's ports to Forwarding. The other device is a Backup and sets all the ports in its VRID VLAN to Blocking.

If a failover occurs, the Backup becomes the new Master and changes all its VRID ports to the Forwarding state.

Other Foundry devices can use the redundant paths provided by the VSRP devices. In this example, three Foundry devices use the redundant paths. A Foundry device that is not itself configured for VSRP but is connected to a Foundry device that is configured for VSRP, is ***VSRP aware***. In this example, the three Foundry devices connected to the VSRP devices are VSRP aware. A Foundry device that is VSRP aware can failover its link to the new Master in sub-second time, by changing the MAC address associated with the redundant path.

When you configure VSRP, make sure each of the non-VSRP Foundry devices connected to the VSRP devices has a separate link to each of the VSRP devices.

Configuration Notes

- VSRP and 802.1Q-n-Q tagging are not supported together on the same device.
- VSRP and Super Aggregated VLANs are not supported together on the same device.

Layer 2 and Layer 3 Redundancy

You can configure VSRP to provide redundancy for Layer 2 only or also for Layer 3.

- Layer 2 only – The Layer 2 links are backed up but specific IP addresses are not backed up.
- Layer 2 and Layer 3 – The Layer 2 links are backed up and a specific IP address is also backed up. Layer 3 VSRP is the same as VRRPE. However, using VSRP provides redundancy at both layers at the same time.

Layer 2 Switches support Layer 2 VSRP only. Layer 3 Switches support Layer 2 and Layer 3 redundancy. You can configure a Layer 3 Switch for either Layer 2 only or Layer 2 and Layer 3. To configure for Layer 3, specify the IP address you are backing up.

NOTE: If you want to provide Layer 3 redundancy only, disable VSRP and use VRRPE.

Master Election and Failover

Each VSRP device advertises its VSRP priority in Hello messages. During Master election, the VSRP device with the highest priority for a given VRID becomes the Master for that VRID. After Master election, the Master sends Hello messages at regular intervals to inform the Backups that the Master is healthy.

If there is a tie for highest VSRP priority, the tie is resolved as follows:

- Layer 2 Switches – The Layer 2 Switch with the higher management IP address becomes the Master.
  - Switches with management IP addresses are preferred over switches without management IP addresses.
  - If neither of the switches has a management IP address, then the switch with the higher MAC address becomes the Master. (VSRP compares the MAC addresses of the ports configured for the VRID, not the base MAC addresses of the switches.)
- Layer 3 Switches – The Layer 3 Switch whose virtual routing interface has a higher IP address becomes the master.

VSRP Failover

Each Backup listens for Hello messages from the Master. The Hello messages indicate that the Master is still available. If the Backups stop receiving Hello messages from the Master, the election process occurs again and the Backup with the highest priority becomes the new Master.

Each Backup waits for a specific period of time, the Dead Interval, to receive a new Hello message from the Master. If the Backup does not receive a Hello message from the Master by the time the Dead Interval expires, the Backup sends a Hello message of its own, which includes the Backup's VSRP priority, to advertise the Backup's intent to become the Master. If there are multiple Backups for the VRID, each Backup sends a Hello message.

When a Backup sends a Hello message announcing its intent to become the Master, the Backup also starts a hold-down timer. During the hold-down time, the Backup listens for a Hello message with a higher priority than its own.

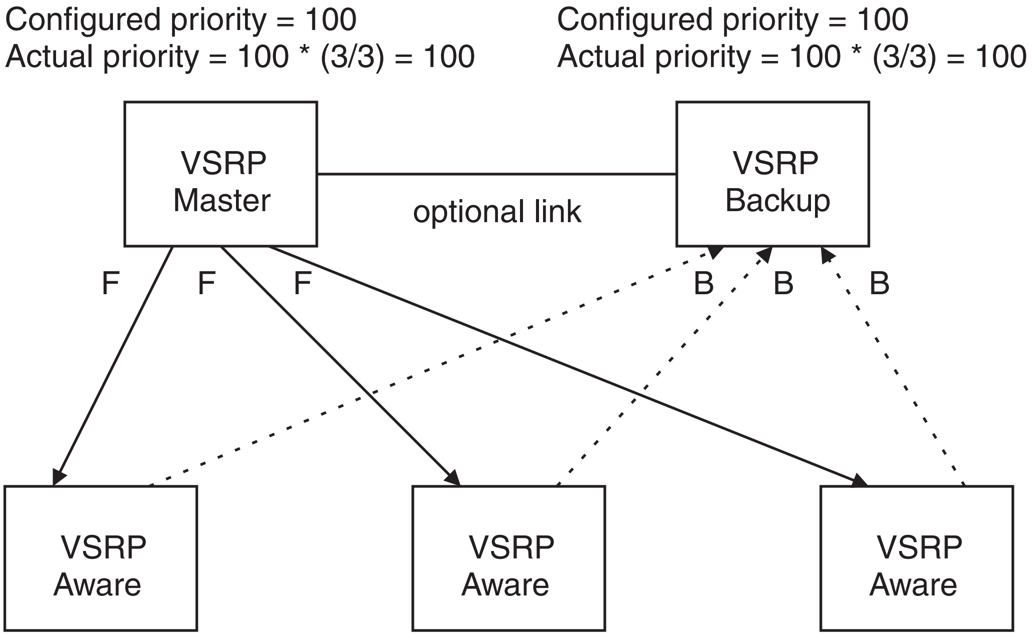
- If the Backup receives a Hello message with a higher priority than its own, the Backup resets its Dead Interval and returns to normal Backup status.
- If the Backup does not receive a Hello message with a higher priority than its own by the time the hold-down timer expires, the Backup becomes the new Master and starts forwarding Layer 2 traffic on all ports.

If you increase the timer scale value, each timer’s value is divided by the scale value. To achieve sub-second failover times, you can change the scale to a value up to 10. This shortens all the VSRP timers to 10 percent of their configured values.

VSRP Priority Calculation

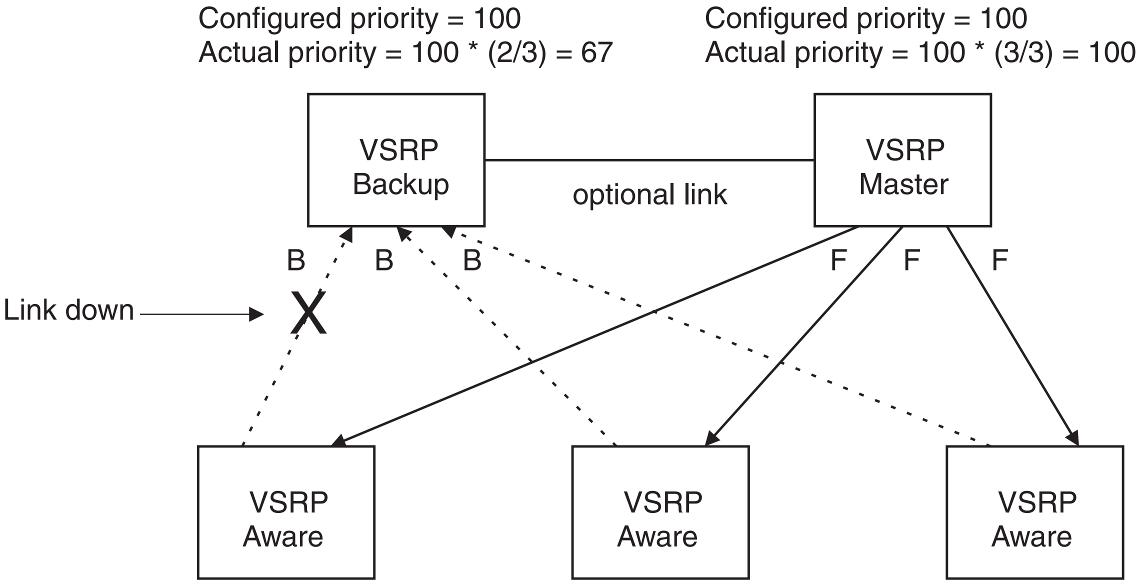
Each VSRP device has a VSRP priority for each VRID and its VLAN. The VRID is used during Master election for the VRID. By default, a device’s VSRP priority is the value configured on the device (which is 100 by default). However, to ensure that a Backup with a high number of up ports for a given VRID is elected, the device reduces the priority if a port in the VRID’s VLAN goes down. For example, if two Backups each have a configured priority of 100, and have three ports in VRID 1 in VLAN 10, each Backup begins with an equal priority, 100. This is shown in Figure 10.12

Figure 10.12 VSRP Priority


Configured priority = 100
Actual priority = 100 * (3/3) = 100
Configured priority = 100
Actual priority = 100 * (3/3) = 100
VSRP Master
optional link
VSRP Backup
F
F
F
B
B
B
VSRP Aware
VSRP Aware
VSRP Aware

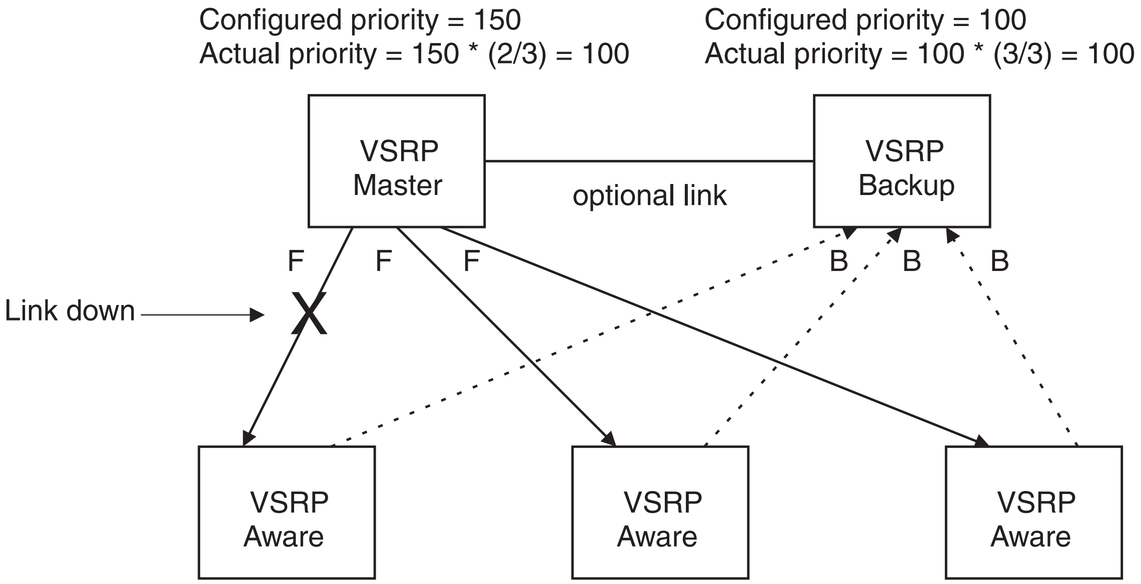
However, if one of the VRID's ports goes down on one of the Backups, that Backup's priority is reduced. If the Master's priority is reduced enough to make the priority lower than a Backup's priority, the VRID fails over to the Backup. Figure 10.13 shows an example.

Figure 10.13 VSRP Priority Recalculation


Configured priority = 100
Actual priority = 100 * (2/3) = 67
Configured priority = 100
Actual priority = 100 * (3/3) = 100
VSRP Backup
optional link
VSRP Master
B
B
B
F
F
F
Link down
X
VSRP Aware
VSRP Aware
VSRP Aware

You can reduce the sensitivity of a VSRP device to failover by increasing its configured VSRP priority. For example, you can increase the configured priority of the VSRP device on the left in Figure 10.13 to 150. In this case, failure of a single link does not cause failover. The link failure caused the priority to be reduced to 100, which is still equal to the priority of the other device. This is shown in Figure 10.14.

Figure 10.14 VSRP Priority Bias

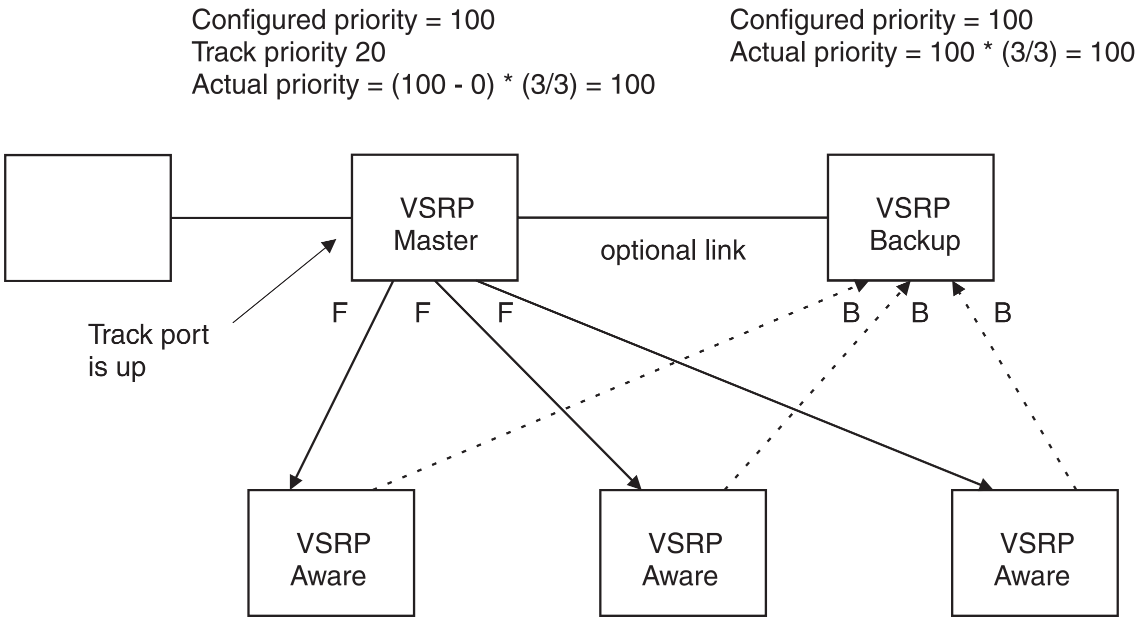

Configured priority = 150
Actual priority = 150 * (2/3) = 100
Configured priority = 100
Actual priority = 100 * (3/3) = 100
VSRP Master
optional link
VSRP Backup
F
F
F
B
B
B
Link down
X
VSRP Aware
VSRP Aware
VSRP Aware

Track Ports

Optionally, you can configure track ports to be included during VSRP priority calculation. In VSRP, a ***track port*** is a port that is not a member of the VRID's VLAN, but whose state is nonetheless considered when the priority is calculated. Typically, a track port represents the exit side of traffic received on the VRID ports. By default, no track ports are configured.

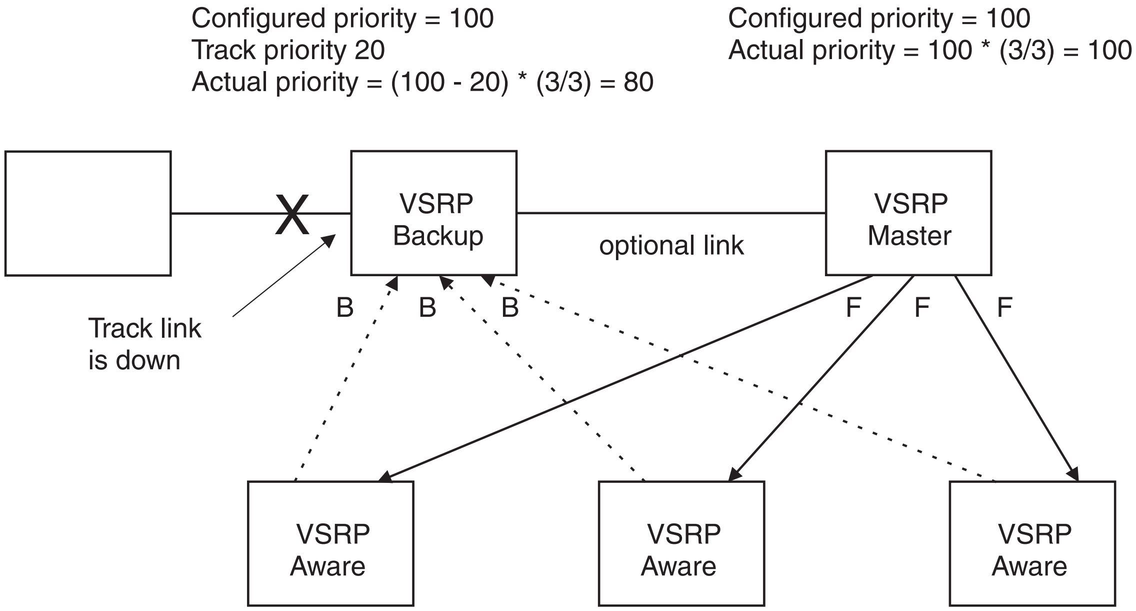
When you configure a track port, you assign a priority value to the port. If the port goes down, VSRP subtracts the track port's priority value from the configured VSRP priority. For example, if the you configure a track port with priority 20 and the configured VSRP priority is 100, the software subtracts 20 from 100 if the track port goes down, resulting in a VSRP priority of 80. The new priority value is used when calculating the VSRP priority. Figure 10.15 shows an example.

Figure 10.15 Track Port Priority


Configured priority = 100
Track priority 20
Actual priority = (100 - 0) * (3/3) = 100
Configured priority = 100
Actual priority = 100 * (3/3) = 100
VSRP Master
optional link
VSRP Backup
Track port is up
F
F
F
B
B
B
VSRP Aware
VSRP Aware
VSRP Aware

In Figure 10.15, the track port is up. SInce the port is up, the track priority does not affect the VSRP priority calculation. If the track port goes down, the track priority does affect VSRP priority calculation, as shown in Figure 10.16.

Figure 10.16 Track Port Priority Subtracted During Priority Calculation


Configured priority = 100
Track priority 20
Actual priority = (100 - 20) * (3/3) = 80
Configured priority = 100
Actual priority = 100 * (3/3) = 100
X
VSRP Backup
optional link
VSRP Master
Track link is down
B
B
B
F
F
F
VSRP Aware
VSRP Aware
VSRP Aware

MAC Address Failover on VSRP-Aware Devices

VSRP-aware devices maintain a record of each VRID and its VLAN. When the device has received a Hello message for a VRID in a given VLAN, the device creates a record for that VRID and VLAN and includes the port number in the record. Each subsequent time the device receives a Hello message for the same VRID and VLAN, the device checks the port number.

- If the port number is the same as the port that previously received a Hello message, the VSRP-aware device assumes that the message came from the same VSRP Master that sent the previous message.
- If the port number does not match, the VSRP-aware device assumes that a VSRP failover has occurred to a new Master, and moves the MAC addresses learned on the previous port to the new port.

The VRID records age out if unused. This can occur if the VSRP-aware device becomes disconnected from the Master. The VSRP-aware device will wait for a Hello message for the period of time equal to the following:

VRID Age = Dead Interval + Hold-down Interval + (3 x Hello Interval)

The values for these timers are determined by the VSRP device sending the Hello messages. If the Master uses the default timer values, the age time for VRID records on the VSRP-aware devices is as follows:

3 + 2 + (3 x 1) = 8 seconds

In this case, if the VSRP-aware device does not receive a new Hello message for a VRID in a given VLAN, on any port, the device assumes the connection to the Master is unavailable and removes the VRID record.

Timer Scale

The VSRP Hello interval, Dead interval, Backup Hello interval, and Hold-down interval timers are individually configurable. You also can easily change all the timers at the same time while preserving the ratios among their values. To do so, change the timer scale. The ***timer scale*** is a value used by the software to calculate the timers. The software divides a timer’s value by the timer scale value. By default, the scale is 1. This means the VSRP timer values are the same as the values in the configuration.

VSRP-Aware Security Features

Platform Support:

- FESX/FSX/FWSX devices running software release 03.0.00 and later
- FGS and FLS devices running software release 03.0.00 and later

This feature protects against unauthorized VSRP hello packets by enabling you to configure VSRP-aware security parameters. Without VSRP-aware security, a VSRP-aware device passively learns the authentication method conveyed by the received VSRP hello packet. The VSRP-aware device then stores the authentication method until it ages out with the aware entry.

The VSRP-aware security feature enables you to:

- Define the specific authentication parameters that a VSRP-aware device will use on a VSRP backup switch. The authentication parameters that you define will not age out.
- Define a list of ports that have authentic VSRP backup switch connections. For ports included in the list, the VSRP-aware switch will process VSRP hello packets using the VSRP-aware security configuration. Conversely, for ports not included in the list, the VSRP-aware switch will not use the VSRP-aware security configuration.

If VSRP hello packets do not meet the acceptance criteria, the VSRP-aware device forwards the packets normally, without any VSRP-aware security processing.

To configure VSRP-Aware Security features, see “Configuring Security Features on a VSRP-Aware Device” on page 10-32.

VSRP Parameters

Table 10.4 lists the VSRP parameters.

Table 10.4: VSRP Parameters

Parameter	Description	Default	See page...
Protocol	VSRP state **Note**: On a Layer 3 Switch, you must disable VSRP to use VRRPE or VRRP.	Enabled	10-31
Virtual Router ID (VRID)	The ID of the virtual switch you are creating by configuring multiple devices as redundant links. You must configure the same VRID on each device that you want to use to back up the links.	None	10-30
Timer scale	The value used by the software to calculate all VSRP timers. Increasing the timer scale value decreases the length of all the VSRP timers equally, without changing the ratio of one timer to another.	1	10-31

Table 10.4: VSRP Parameters (Continued)

Parameter	Description	Default	See page...
Interface Parameters			
Authentication type	The type of authentication the VSRP devices use to validate VSRP packets. On Layer 3 Switches, the authentication type must match the authentication type the VRID's port uses with other routing protocols such as OSPF. • No authentication – The interfaces do not use authentication. This is the VRRP default. • Simple – The interface uses a simple text-string as a password in packets sent on the interface. If the interface uses simple password authentication, the VRID configured on the interface must use the same authentication type and the same password. **Note**: MD5 is not supported.	No authentication	10-32
VSRP-Aware Security Parameters			
VSRP-Aware Authentication type	The type of authentication the VSRP-aware devices will use on a VSRP backup switch. • No authentication – The device does not accept incoming packets that have authentication strings. • Simple – The device uses a simple text-string as the authentication string for accepting incoming packets.	Not configured	10-32
VRID Parameters			
VSRP device type	Whether the device is a VSRP Backup for the VRID. All VSRP devices for a given VRID are Backups.	Not configured	10-30
VSRP ports	The ports in the VRID's VLAN that you want to use as VRID interfaces. You can selectively exclude individual ports from VSRP while allowing them to remain in the VLAN.	All ports in the VRID's VLAN	10-33
VRID IP address	A gateway address you are backing up. Configuring an IP address provides VRRPE Layer 3 redundancy in addition to VSRP LAyer 2 redundancy. The VRID IP address must be in the same subnet as a real IP address configured on the VSRP interface, but cannot be the same as a real IP address configured on the interface. **Note**: This parameter is valid only on Layer 3 Switches.	None	10-33

Table 10.4: VSRP Parameters (Continued)

Parameter	Description	Default	See page...
Backup priority	A numeric value that determines a Backup's preferability for becoming the Master for the VRID. During negotiation, the device with the highest priority becomes the Master. In VSRP, all devices are Backups and have the same priority by default. If two or more Backups are tied with the highest priority, the Backup with the highest IP address becomes the Master for the VRID.	100 for all Backups	10-34
Preference of timer source	When you save a Backup's configuration, the software can save the configured VSRP timer values or the VSRP timer values received from the Master. Saving the current timer values instead of the configured ones helps ensure consistent timer usage for all the VRID's devices. **Note**: The Backup always gets its timer scale value from the Master.	Configured timer values are saved	10-34
Time-to-Live (TTL)	The maximum number of hops a VSRP Hello packet can traverse before being dropped. You can specify from 1 – 255.	2	10-35
Hello interval	The amount of time between Hello messages from the Master to the Backups for a given VRID. The interval can be from 1 – 84 seconds.	One second	10-35
Dead interval	The amount of time a Backup waits for a Hello message from the Master for the VRID before determining that the Master is no longer active. If the Master does not send a Hello message before the dead interval expires, the Backups negotiate (compare priorities) to select a new Master for the VRID.	Three times the Hello Interval	10-35
Backup Hello state and interval	The amount of time between Hello messages from a Backup to the Master. The message interval can be from 60 – 3600 seconds. You must enable the Backup to send the messages. The messages are disabled by default on Backups. The current Master sends Hello messages by default.	Disabled 60 seconds when enabled	10-36
Hold-down interval	The amount of time a Backup that has sent a Hello packet announcing its intent to become Master waits before beginning to forward traffic for the VRID. The hold-down interval prevents Layer 2 loops from occurring during VSRP's rapid failover. The interval can from 1 – 84 seconds.	2 seconds	10-36

Table 10.4: VSRP Parameters (Continued)

Parameter	Description	Default	See page...
Track priority	A VSRP priority value assigned to the tracked port(s). If a tracked port's link goes down, the VRID port's VSRP priority is reduced by the amount of the tracked port's priority.	5	10-36
Track port	A track port is a port or virtual routing interface that is outside the VRID but whose link state is tracked by the VRID. Typically, the tracked interface represents the other side of VRID traffic flow through the device. If the link for a tracked interface goes down, the VSRP priority of the VRID interface is changed, causing the devices to renegotiate for Master.	None	10-37
Backup preempt mode	Prevents a Backup with a higher VSRP priority from taking control of the VRID from another Backup that has a lower priority but has already assumed control of the VRID.	Enabled	10-37
VRID active state	The active state of the VSRP VRID.	Disabled	10-30
RIP Parameters			
Suppression of RIP advertisements	A Layer 3 Switch that is running RIP normally advertises routes to a backed up VRID even when the Layer 3 Switch is not currently the active Layer 3 Switch for the VRID. Suppression of these advertisements helps ensure that other Layer 3 Switches do not receive invalid route paths for the VRID. **Note**: This parameter is valid only on Layer 3 Switches.	Disabled (routes are advertised)	10-37

Configuring Basic VSRP Parameters

To configure VSRP, perform the following required tasks:

- Configure a port-based VLAN containing the ports for which you want to provide VSRP service.

  NOTE: If you already have a port-based VLAN but only want to use VSRP on a sub-set of the VLANs ports, you can selectively remove ports from VSRP service in the VLAN. See "Removing a Port from the VRID's VLAN" on page 10-33.

- Configure a VRID.
  - Specify that the device is a backup. Since VSRP, like VRRPE, does not have an "owner", all VSRP devices are backups. The active device for a VRID is elected based on the VRID priority, which is configurable.
  - Activate the VRID.

The following example shows a simple VSRP configuration.

```
FastIron(config)#vlan 200
FastIron(config-vlan-200)#tag ethernet 1/1 to 1/8
FastIron(config-vlan-200)#vsrp vrid 1
FastIron(config-vlan-200-vrid-1)#backup
FastIron(config-vlan-200-vrid-1)#activate
```

Syntax: [no] vsrp vrid <num>

The <num> parameter specifies the VRID and can be from 1 – 255.

Syntax: [no] backup [priority <value>] [track-priority <value>]

This command is required. In VSRP, all devices on which a VRID are configured are Backups. The Master is then elected based on the VSRP priority of each device. There is no “owner” device as there is in VRRP.

For information about the command’s optional parameters, see the following:

- “Changing the Backup Priority” on page 10-34
- “Changing the Default Track Priority” on page 10-36

Syntax: [no] activate

or

Syntax: enable | disable

Configuring Optional VSRP Parameters

The following sections describe how to configure optional VSRP parameters.

Disabling or Re-Enabling VSRP

VSRP is enabled by default on Layer 2 Switches and Layer 3 Switches. On a Layer 3 Switch, if you want to use VRRP or VRRPE for Layer 3 redundancy instead of VSRP, you need to disable VSRP first. To do so, enter the following command at the global CONFIG level:

```
FastIron(config)#no router vsrp
router vsrp is disabled. All vsrp config data will be lost when writing to flash
```

To re-enable the protocol, enter the following command:

```
FastIron(config)#router vsrp
```

Syntax: [no] router vsrp

Since VRRP and VRRPE do not apply to Layer 2 Switches, there is no need to disable VSRP and there is no command to do so. The protocol is always enabled.

Changing the Timer Scale

To achieve sub-second failover times, you can shorten the duration of all scale timers for VSRP, VRRP, and VRRP-E by adjusting the timer scale. The ***timer scale*** is a value used by the software to calculate the timers. By default, the scale value is 1. If you increase the timer scale, each timer’s value is divided by the scale value. Using the timer scale to adjust timer values enables you to easily change all the timers while preserving the ratios among their values. Here is an example.

Timer	Timer Scale	Timer Value
Hello interval	1	1 second
	2	0.5 seconds
Dead interval	1	3 seconds
	2	1.5 seconds

Timer	Timer Scale	Timer Value
Backup Hello interval	1	60 seconds
	2	30 seconds
Hold-down interval	1	2 seconds
	2	1 second

If you configure the device to receive its timer values from the Master, the Backup also receives the timer scale value from the Master.

NOTE: The Backups always use the value of the timer scale received from the Master, regardless of whether the timer values that are saved in the configuration are the values configured on the Backup or the values received from the Master.

To change the timer scale, enter a command such as the following at the global CONFIG level of the CLI:

```
FastIron SuperX Router(config)# scale-timer 2
```

This command changes the scale to 2. All VSRP, VRRP, and VRRP-E timer values will be divided by 2.

Syntax: [no] scale-timer <num>

The <num> parameter specifies the multiplier. You can specify a timer scale from 1 – 10.

Configuring Authentication

If the interfaces on which you configure the VRID use authentication, the VSRP packets on those interfaces also must use the same authentication. VSRP supports the following authentication types:

- No authentication – The interfaces do not use authentication. This is the default.
- Simple – The interfaces use a simple text-string as a password in packets sent on the interface. If the interfaces use simple password authentication, the VRID configured on the interfaces must use the same authentication type and the same password.

To configure a simple password, enter a command such as the following at the interface configuration level:

```
FastIron(config-if-1/6)#ip vsrp auth-type simple-text-auth ourpword
```

This command configures the simple text password “ourpword”.

Syntax: [no] ip vsrp auth-type no-auth | simple-text-auth <auth-data>

The **auth-type no-auth** parameter indicates that the VRID and the interface it is configured on do not use authentication.

The **auth-type simple-text-auth** <auth-data> parameter indicates that the VRID and the interface it is configured on use a simple text password for authentication. The <auth-data> value is the password. If you use this parameter, make sure all interfaces on all the devices supporting this VRID are configured for simple password authentication and use the same password.

Configuring Security Features on a VSRP-Aware Device

Platform Support:

- FESX/FSX/FWSX devices running software release 03.0.00 and later
- FGS and FLS devices running software release 03.0.00 and later

This section shows how to configure security features on a VSRP-aware device. For an overview of this feature, see “VSRP-Aware Security Features” on page 10-27.

Specifying an Authentication String for VSRP Hello Packets

The following configuration defines **pri-key** as the authentication string for accepting incoming VSRP hello packets. In this example, the VSRP-aware device will accept all incoming packets that have this authorization string.

```
FastIron(config)#vlan 10
FastIron(config-vlan-10)#vsrp-aware vrid 3 simple-text-auth pri-key
```

Syntax: vsrp-aware vrid <vrid number> simple text auth <string>

Specifying no Authentication for VSRP Hello Packets

The following configuration specifies no authentication as the preferred VSRP-aware security method. In this case, the VSRP device will not accept incoming packets that have authentication strings.

```
FastIron(config)#vlan 10
FastIron(config-vlan-10)#vsrp-aware vrid 2 no-auth
```

Syntax: vsrp-aware vrid <vrid number> no-auth

The following configuration specifies no authentication for VSRP hello packets received on ports 1/1, 1/2, 1/3, and 1/4 in VRID 4. For these ports, the VSRP device will not accept incoming packets that have authentication strings.

```
FastIron(config)#vlan 10
FastIron(config-vlan-10)#vsrp-aware vrid 4 no-auth port-list ethe 1/1 to 1/4
```

Syntax: vsrp-aware vrid <vrid number> no-auth port-list <port range>

<vrid number> is a valid VRID (from 1 to 255).

no-auth specifies no authentication as the preferred VSRP-aware security method. The VSRP device will not accept incoming packets that have authentication strings.

simple-text-auth <string> specifies the authentication string for accepting VSRP hello packets, where <string> can be up to 8 characters.

port-list <port range> specifies the range of ports to include in the configuration.

Removing a Port from the VRID's VLAN

By default, all the ports in the VLAN on which you configure a VRID are interfaces for the VRID. You can remove a port from the VRID while allowing it to remain in the VLAN.

Removing a port is useful in the following cases:

- There is no risk of a loop occurring, such as when the port is attached directly to an end host.
- You plan to use a port in an MRP ring.

To remove a port from a VRID, enter a command such as the following at the configuration level for the VRID:

```
FastIron(config-vlan-200-vrid-1)#no include-port ethernet 1/2
```

Syntax: [no] include-port ethernet [<slotnum>/]<portnum>

The <slotnum> parameter is required on chassis devices.

The <portnum> parameter specifies the port you are removing from the VRID. The port remains in the VLAN but its forwarding state is not controlled by VSRP. If you are configuring a chassis device, specify the slot number as well as the port number (<slotnum>/<portnum>).

Configuring a VRID IP Address

If you are configuring a Layer 3 Switch for VSRP, you can specify an IP address to back up. When you specify an IP address, VSRP provides redundancy for the address. This is useful if you want to back up the gateway address used by hosts attached to the VSRP Backups.

VSRP does not require you to specify an IP address. If you do not specify an address, VSRP provides Layer 2 redundancy. If you do specify an address, VSRP provides Layer 2 and Layer 3 redundancy.

The Layer 3 redundancy support is the same as VRRPE support. For information, see the chapter “Configuring VRRP and VRRPE” on page 37-1.

NOTE: The VRID IP address must be in the same subnet as a real IP address configured on the VSRP interface, but cannot be the same as a real IP address configured on the interface.

NOTE: Failover applies to both Layer 2 and Layer 3.

To specify an IP address to back up, enter a command such as the following at the configuration level for the VRID:

```
FastIron(config-vlan-200-vrid-1)#ip-address 10.10.10.1
```

Syntax: [no] ip-address <ip-addr>

or

Syntax: [no] ip address <ip-addr>

Changing the Backup Priority

When you enter the backup command to configure the device as a VSRP Backup for the VRID, you also can change the backup priority and the track priority.

- The backup priority is used for election of the Master. The VSRP Backup with the highest priority value for the VRID is elected as the Master for that VRID. The default priority is 100. If two or more Backups are tied with the highest priority, the Backup with the highest IP address becomes the Master for the VRID.
- The track priority is used with the track port feature. See “VSRP Priority Calculation” on page 10-23 and “Changing the Default Track Priority” on page 10-36.

To change the backup priority, enter a command such as the following at the configuration level for the VRID:

```
FastIron(config-vlan-200-vrid-1)#backup priority 75
```

Syntax: [no] backup [priority <value>] [track-priority <value>]

The **priority** <value> parameter specifies the VRRP priority for this interface and VRID. You can specify a value from 3 – 254. The default is 100.

For a description of the **track-priority** <value> parameter, see “Changing the Default Track Priority” on page 10-36.

Saving the Timer Values Received from the Master

The Hello messages sent by a VRID’s master contain the VRID values for the following VSRP timers:

- Hello interval
- Dead interval
- Backup Hello interval
- Hold-down interval

By default, each Backup saves the configured timer values to its startup-config file when you save the device’s configuration.

You can configure a Backup to instead save the current timer values received from the Master when you save the configuration. Saving the current timer values instead of the configured ones helps ensure consistent timer usage for all the VRID’s devices.

NOTE: The Backups always use the value of the timer scale received from the Master, regardless of whether the timer values that are saved in the configuration are the values configured on the Backup or the values received from the Master.

To configure a Backup to save the VSRP timer values received from the Master instead of the timer values configured on the Backup, enter the following command:

```
FastIron(config-vlan-200-vrid-1)#save-current-values
```

Syntax: [no] save-current-values

Changing the Time-To-Live (TTL)

A VSRP Hello packet's TTL specifies how many hops the packet can traverse before being dropped. A hop can be a Layer 3 Switch or a Layer 2 Switch. You can specify from 1 – 255. The default TTL is 2. When a VSRP device (Master or Backup) sends a VSRP HEllo packet, the device subtracts one from the TTL. Thus, if the TTL is 2, the device that originates the Hello packet sends it out with a TTL of 1. Each subsequent device that receives the packet also subtracts one from the packet's TTL. When the packet has a TTL of 1, the receiving device subtracts 1 and then drops the packet because the TTL is zero.

NOTE: An MRP ring is considered to be a single hop, regardless of the number of nodes in the ring.

To change the TTL for a VRID, enter a command such as the following at the configuration level for the VRID:

```
FastIron(config-vlan-200-vrid-1)#initial-ttl 5
```

Syntax: [no] initial-ttl <num>

The <num> parameter specifies the TTL and can be from 1 – 255. The default TTL is 2.

Changing the Hello Interval

The Master periodically sends Hello messages to the Backups. To change the Hello interval, enter a command such as the following at the configuration level for the VRID:

```
FastIron(config-vlan-200-vrid-1)#hello-interval 10
```

Syntax: [no] hello-interval <num>

The <num> parameter specifies the interval and can be from 1 – 84 seconds. The default is 1 second.

NOTE: The default Dead interval is three times the Hello interval plus one-half second. Generally, if you change the Hello interval, you also should change the Dead interval on the Backups.

NOTE: If you change the timer scale, the change affects the actual number of seconds.

Changing the Dead Interval

The Dead interval is the number of seconds a Backup waits for a Hello message from the Master before determining that the Master is dead. The default is 3 seconds. This is three times the default Hello interval.

To change the Dead interval, enter a command such as the following at the configuration level for the VRID:

```
FastIron(config-vlan-200-vrid-1)#dead-interval 30
```

Syntax: [no] dead-interval <num>

The <num> parameter specifies the interval and can be from 1 – 84 seconds. The default is 3 seconds.

NOTE: If you change the timer scale, the change affects the actual number of seconds.

Changing the Backup Hello State and Interval

By default, Backups do not send Hello messages to advertise themselves to the Master. You can enable these messages if desired and also change the message interval.

To enable a Backup to send Hello messages to the Master, enter a command such as the following at the configuration level for the VRID:

```
FastIron(config-vlan-200-vrid-1)#advertise backup
```

Syntax: [no] advertise backup

When a Backup is enabled to send Hello messages, the Backup sends a Hello message to the Master every 60 seconds by default. You can change the interval to be up to 3600 seconds.

To change the Backup Hello interval, enter a command such as the following at the configuration level for the VRID:

```
FastIron(config-vlan-200-vrid-1)#backup-hello-interval 180
```

Syntax: [no] backup-hello-interval <num>

The <num> parameter specifies the message interval and can be from 60 – 3600 seconds. The default is 60 seconds.

NOTE: If you change the timer scale, the change affects the actual number of seconds.

Changing the Hold-Down Interval

The hold-down interval prevents Layer 2 loops from occurring during failover, by delaying the new Master from forwarding traffic long enough to ensure that the failed Master is really unavailable.

To change the Hold-down interval, enter a command such as the following at the configuration level for the VRID:

```
FastIron(config-vlan-200-vrid-1)#hold-down-interval 4
```

Syntax: [no] hold-down-interval <num>

The <num> parameter specifies the hold-down interval and can be from 1 – 84 seconds. The default is 2 seconds.

NOTE: If you change the timer scale, the change affects the actual number of seconds.

Changing the Default Track Priority

When you configure a VRID to track the link state of other interfaces, if one of the tracked interface goes down, the software changes the VSRP priority of the VRID interface.

The software reduces the VRID priority by the amount of the priority of the tracked interface that went down. For example, if the VSRP interface's priority is 100 and a tracked interface with track priority 60 goes down, the software changes the VSRP interface's priority to 40. If another tracked interface goes down, the software reduces the VRID's priority again, by the amount of the tracked interface's track priority.

The default track priority for all track ports is 1. You can change the default track priority or override the default for an individual track port.

- To change the default track priority, use the **backup track-priority** command, described below.
- To override the default track priority for a specific track port, use the **track-port** command. See "Specifying a Track Port" on page 10-37.

To change the track priority, enter a command such as the following at the configuration level for the VRID:

```
FastIron(config-vlan-200-vrid-1)#backup track-priority 2
```

Syntax: [no] backup [priority <value>] [track-priority <value>]

Specifying a Track Port

You can configure the VRID on one interface to track the link state of another interface on the device. This capability is useful for tracking the state of the exit interface for the path for which the VRID is providing redundancy. See "VSRP Priority Calculation" on page 10-23.

To configure a VRID to track an interface, enter a command such as the following at the configuration level for the VRID:

```
FastIron(config-vlan-200-vrid-1)#track-port e 2/4
```

Syntax: [no] track-port ethernet [<slotnum>/]<portnum> | ve <num> [priority <num>]

The **priority** <num> parameter changes the VSRP priority of the interface. If this interface goes down, the VRID's VSRP priority is reduced by the amount of the track port priority you specify here.

NOTE: The priority <num> option changes the priority of the specified interface, overriding the default track port priority. To change the default track port priority, use the **backup track-priority** <num> command.

Disabling or Re-Enabling Backup Pre-Emption

By default, a Backup that has a higher priority than another Backup that has become the Master can preempt the Master, and take over the role of Master. If you want to prevent this behavior, disable preemption.

Preemption applies only to Backups and takes effect only when the Master has failed and a Backup has assumed ownership of the VRID. The feature prevents a Backup with a higher priority from taking over as Master from another Backup that has a lower priority but has already become the Master of the VRID.

Preemption is especially useful for preventing flapping in situations where there are multiple Backups and a Backup with a lower priority than another Backup has assumed ownership, because the Backup with the higher priority was unavailable when ownership changed.

If you enable the non-preempt mode (thus disabling the preemption feature) on all the Backups, the Backup that becomes the Master following the disappearance of the Master continues to be the Master. The new Master is not preempted.

To disable preemption on a Backup, enter a command such as the following at the configuration level for the VRID:

```
FastIron(config-vlan-200-vrid-1)#non-preempt-mode
```

Syntax: [no] non-preempt-mode

Suppressing RIP Advertisement from Backups

Normally, for Layer 3 a VSRP Backup includes route information for a backed up IP address in RIP advertisements. As a result, other Layer 3 Switches receive multiple paths for the backed up interface and might sometimes unsuccessfully use the path to the Backup rather than the path to the Master.

You can prevent the Backups from advertising route information for the backed up interface by enabling suppression of the advertisements.

NOTE: This parameter applies only if you specified an IP address to back up and is valid only on Layer 3 Switches.

To suppress RIP advertisements, enter the following commands:

```
Router2(config)#router rip
Router2(config-rip-router)#use-vrrp-path
```

Syntax: [no] use-vrrp-path

VSRP-Aware Interoperability

The **vsrp-aware tc-vlan-flush** command should be used in network configurations in which the FGS or FLS switch operates as the VSRP-Aware device connecting to a Foundry BigIron RX, NetIron XMR, or NetIron MLX configured as a VSRP Master.

The command is available at the VLAN level, and is issued per a specific VRID, as shown here for VRID 11:

```
FastIron(config-vlan-10)#vsrp-aware vrid 11 tc-vlan-flush
```

Syntax: vsrp-aware vrid <num> tc-vlan-flush

When this command is enabled, MAC addresses will be flushed at the VLAN level, instead of at the port level. MAC addresses will be flushed for every topology change (TC) received on the VSRP-aware ports.

When this command is enabled, the results of the **show vsrp-aware vlan** command resemble the following:

```
FastIron(config-vlan-10)#vsrp-aware vrid 11 tc-vlan-flush
FastIron(config-vlan-10)#show vsrp aware vlan 10
Aware Port Listing
   VLAN ID VRID Last Port Auth Type       Mac-Flush  Age
    10       11 N/A   no-auth  Configured Enabled    00:00:00.0
```

Displaying VSRP Information

You can display the following VSRP information:

- Configuration information and current parameter values for a VRID or VLAN
- The interfaces on a VSRP-aware device that are active for the VRID

Displaying VRID Information

To display VSRP information, enter the following command:

```
FastIron#show vsrp vrid 1
Total number of VSRP routers defined: 2
VLAN 200
 auth-type no authentication
 VRID 1
  State        Administrative-status Advertise-backup Preempt-mode save-current
  standby      enabled               disabled         true         false

  Parameter        Configured Current    Unit
  priority                100     80     (100-0)*(4.0/5.0)
  hello-interval            1      1     sec/1
  dead-interval             3      3     sec/1
  hold-interval             3      3     sec/1
  initial-ttl               2      2     hops

  next hello sent in 00:00:00.8
  Member ports:      ethe 1/1 to 1/5
  Operational ports: ethe 1/1 to 1/4
  Forwarding ports:  ethe 1/1 to 1/4
```

Syntax: show vsrp [vrid <num> | vlan <vlan-id>]

This display shows the following information when you use the **vrid** <num> or **vlan** <vlan-id> parameter. For information about the display when you use the **aware** parameter, see "Displaying the Active Interfaces for a VRID" on page 10-40.

Table 10.5: CLI Display of VSRP VRID or VLAN Information

This Field...	Displays...
Total number of VSRP routers defined	The total number of VRIDs configured on this device.
VLAN	The VLAN on which VSRP is configured.
auth-type	The authentication type in effect on the ports in the VSRP VLAN.
VRID parameters	
VRID	The VRID for which the following information is displayed.
state	This device's VSRP state for the VRID. The state can be one of the following: • initialize – The VRID is not enabled (activated). If the state remains "initialize" after you activate the VRID, make sure that the VRID is also configured on the other routers and that the routers can communicate with each other. **Note**: If the state is "initialize" and the mode is incomplete, make sure you have specified the IP address for the VRID. • standby – This device is a Backup for the VRID. • master – This device is the Master for the VRID.
Administrative-status	The administrative status of the VRID. The administrative status can be one of the following: • disabled – The VRID is configured on the interface but VSRP or VRRPE has not been activated on the interface. • enabled – VSRP has been activated on the interface.
Advertise-backup	Whether the device is enabled to send VSRP Hello messages when it is a Backup. This field can have one of the following values: • disabled – The device does not send Hello messages when it is a Backup. • enabled – The device does send Hello messages when it is a Backup.
Preempt-mode	Whether the device can be pre-empted by a device with a higher VSRP priority after this device becomes the Master. This field can have one of the following values: • disabled – The device cannot be pre-empted. • enabled – The device can be pre-empted.
save-current	The source of VSRP timer values preferred when you save the configuration. This field can have one of the following values: • false – The timer values configured on this device are saved. • true – The timer values most recently received from the Master are saved instead of the locally configured values.

Table 10.5: CLI Display of VSRP VRID or VLAN Information (Continued)

This Field...	Displays...
Note: For the following fields: • Configured – indicates the parameter value configured on this device. • Current – indicates the parameter value received from the Master. • Unit – indicates the formula used tor calculating the VSRP priority and the timer scales in effect for the VSRP timers. A timer's true value is the value listed in the Configured or Current field divided by the scale value.	
priority	The device's preferability for becoming the Master for the VRID. During negotiation, the Backup with the highest priority becomes the Master. If two or more Backups are tied with the highest priority, the Backup interface with the highest IP address becomes the Master for the VRID.
hello-interval	The number of seconds between Hello messages from the Master to the Backups for a given VRID.
dead-interval	The configured value for the dead interval. The dead interval is the number of seconds a Backup waits for a Hello message from the Master for the VRID before determining that the Master is no longer active. If the Master does not send a Hello message before the dead interval expires, the Backups negotiate (compare priorities) to select a new Master for the VRID. **Note**: If the value is 0, then you have not configured this parameter.
hold-interval	The number of seconds a Backup that intends to become the Master will wait before actually beginning to forward Layer 2 traffic for the VRID. If the Backup receives a Hello message with a higher priority than its own before the hold-down interval expires, the Backup remains in the Backup state and does not become the new Master.
initial-ttl	The number of hops a Hello message can traverse after leaving the device before the Hello message is dropped. **Note**: An MRP ring counts as one hop, regardless of the number of nodes in the ring.
next hello sent in	The amount of time until the Master's dead interval expires. If the Backup does not receive a Hello message from the Master by the time the interval expires, either the IP address listed for the Master will change to the IP address of the new Master, or this Layer 3 Switch itself will become the Master. **Note**: This field applies only when this device is a Backup.
Member ports	The ports in the VRID.
Operational ports	The member ports that are currently up.
Forwarding ports	The member ports that are currently in the Forwarding state. Ports that are forwarding on the Master are listed. Ports on the Standby, which are in the Blocking state, are not listed.

Displaying the Active Interfaces for a VRID

On a VSRP-aware device, you can display VLAN and port information for the connections to the VSRP devices (Master and Backups).

To display the active VRID interfaces, enter the following command on the VSRP-aware device:

```
FastIron#show vsrp aware
```

```
Aware port listing
VLAN ID  VRID  Last Port
100      1     3/2
200      2     4/1
```

Syntax: show vsrp aware

This display shows the following information when you use the **aware** parameter. For information about the display when you use the **vrid** <num> or **vlan** <vlan-id> parameter, see "Displaying VRID Information" on page 10-38.

Table 10.6: CLI Display of VSRP-Aware Information

This Field...	Displays...
VLAN ID	The VLAN that contains the VSRP-aware device's connection with the VSRP Master and Backups.
VRID	The VRID.
Last Port	The most recent active port connection to the VRID. This is the port connected to the current Master. If a failover occurs, the VSRP-aware device changes the port to the port connected to the new Master. The VSRP-aware device uses this port to send and receive data through the backed up node.

VSRP Fast Start

VSRP fast start allows non-Foundry or non-VSRP aware devices that are connected to a Foundry device that is the VSRP Master to quickly switchover to the new Master when a VSRP failover occurs

This feature causes the port on a VSRP Master to restart when a VSRP failover occurs. When the port shuts down at the start of the restart, ports on the non-VSRP aware devices that are connected to the VSRP Master flush the MAC address they have learned for the VSRP master. After a specified time, the port on the previous VSRP Master (which now becomes the Backup) returns back online. Ports on the non-VSRP aware devices switch over to the new Master and learn its MAC address.

Configuring VSRP Fast Start

The VSRP fast start feature can be enabled on a VSRP-configured Foundry device, either on the VLAN to which the VRID of the VSRP-configured device belongs (globally) or on a port that belongs to the VRID.

To globally configure a VSRP-configured device to shut down its ports when a failover occurs, then restart after five seconds, enter the following command:

```
FastIron(configure)#vlan 100
FastIron(configure-vlan-100)#vsrp vrid 1
FastIron(configure-vlan-100-vrid-1)#restart-ports 5
```

Syntax: [no] restart-ports <seconds>

This command shuts down all the ports that belong to the VLAN when a failover occurs. All the ports will have the specified VRID.

To configure a single port on a VSRP-configured device to shut down when a failover occurs, then restart after a period of time, enter the following command:

```
FastIron(configure)#interface ethernet 1/1
FastIron(configure-if-1/1)#vsrp restart-port 5
```

Syntax: [no] vsrp restart-port <seconds>

In both commands, the <seconds> parameter instructs the VSRP Master to shut down its port for the specified number of seconds before it starts back up. Enter a value between 1 – 120 seconds. The default is 1 second.

Displaying Ports that Have the VSRP Fast Start Feature Enabled

The **show vsrp vrid** command shows the ports on which the VSRP fast start feature is enabled.

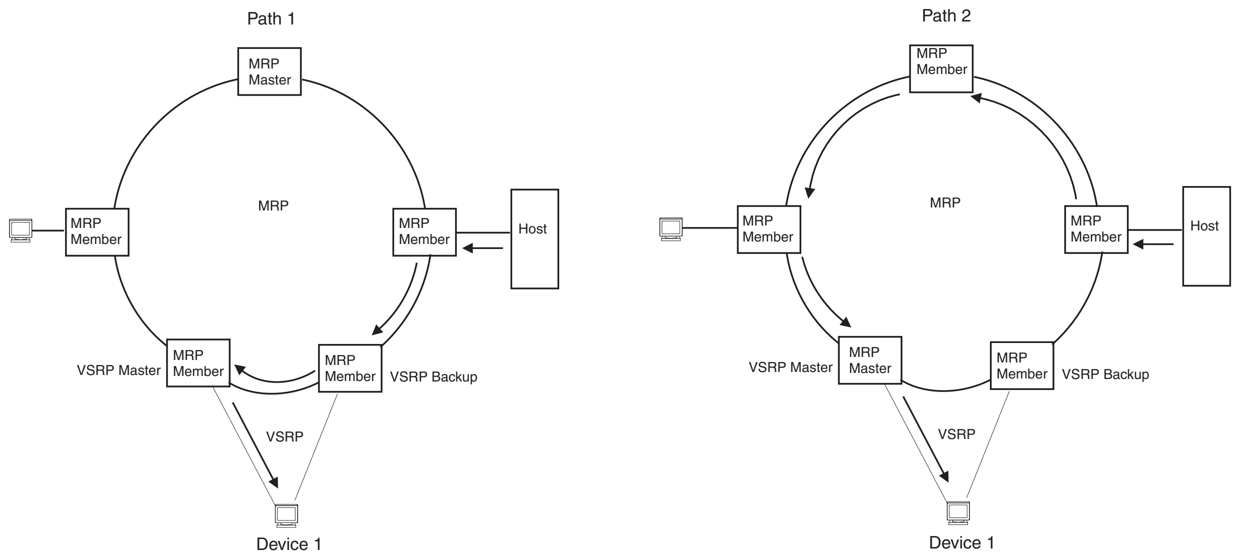
```
FastIron#show vsrp vrid 100

VLAN 100
  auth-type no authentication
  VRID 100
  ========
   State      Administrative-status Advertise-backup Preempt-mode save-current
   master     enabled               disabled         true         false
   Parameter        Configured Current    Unit/Formula
   priority         100        50         (100-0)*(2.0/4.0)
   hello-interval   1          1          sec/1
   dead-interval    3          3          sec/1
   hold-interval    3          3          sec/1
   initial-ttl      2          2          hops

   next hello sent in 00:00:00.3
   Member ports:      ethe 2/5 to 2/8
   Operational ports: ethe 2/5 ethe 2/8
   Forwarding ports:  ethe 2/5 ethe 2/8
   Restart ports:     2/5(1) 2/6(1) 2/7(1) 2/8(1)
```

The "Restart ports:" line lists the ports that have the VSRP fast start enabled, and the downtime for each port. See Table 10.5 on page 10-39 to interpret the remaining information on the display.

VSRP and MRP Signaling

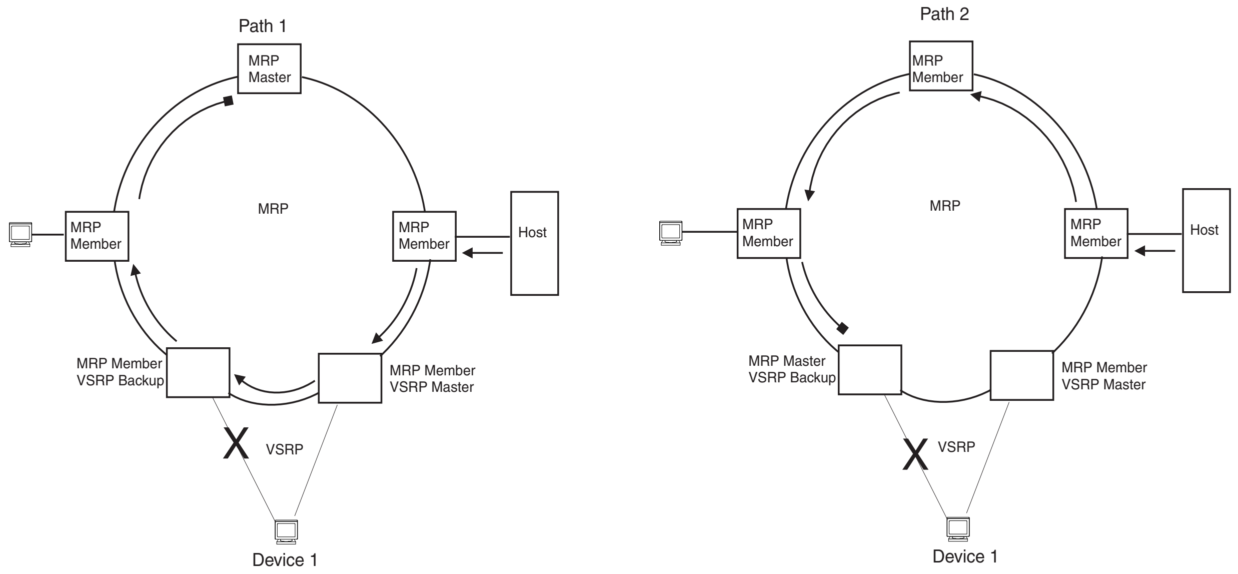
A device may connect to an MRP ring via VSRP to provide a redundant path between the device and the MRP ring. VSRP and MRP signaling ensures rapid failover by flushing MAC addresses appropriately. The host on the MRP ring learns the MAC addresses of all devices on the MRP ring and VSRP link. From these MAC addresses, the host creates a MAC database (table), which is used to establish a data path from the host to a VSRP-linked device. Figure 10.17 below shows two possible data paths from the host to Device 1.

Figure 10.17 Two Data Paths from Host on an MRP Ring to a VSRP-Linked Device



If a VSRP failover from master to backup occurs, VSRP needs to inform MRP of the topology change; otherwise, data from the host continues along the obsolete learned path and never reach the VSRP-linked device, as shown in Figure 10.18.

Figure 10.18 VSRP on MRP Rings that Failed Over

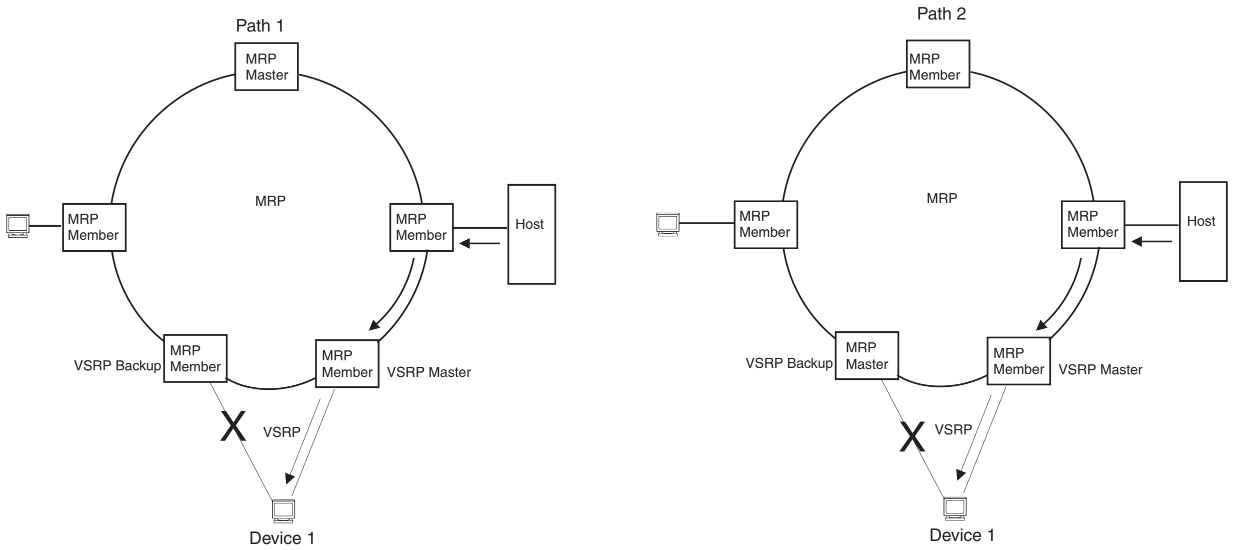


A signaling process for the interaction between VSRP and MRP ensures that MRP is informed of the topology change and achieves convergence rapidly. When a VSRP node fails, a new VSRP master is selected. The new VSRP master finds all MRP instances impacted by the failover. Then each MRP instance does the following:

- The MRP node sends out an MRP PDU with the mac-flush flag set three times on the MRP ring.
- The MRP node that receives this MRP PDU empties all the MAC entries from its interfaces that participate on the MRP ring.
- The MRP node then forwards the MRP PDU with the mac-flush flag set to the next MRP node that is in forwarding state.

The process continues until the Master MRP node's secondary (blocking) interface blocks the packet. Once the MAC address entries have been flushed, the MAC table can be rebuilt for the new path from the host to the VSRP-linked device (Figure 10.19).

Figure 10.19 New Path Established



There are no CLI commands used to configure this process.

Chapter 11

Configuring Power Over Ethernet

This chapter provides an overview of Power over Ethernet (POE) and describes how to enable or disable POE and how to configure POE parameters using CLI commands.

NOTE: This chapter applies to POE devices only.

Power over Ethernet Overview

This section provides an overview of the requirements for delivering power over the LAN, as defined by the Institute of Electrical and Electronics Engineers Inc. (IEEE) in the 802.3af specification.

Foundry's POE devices provide Power over Ethernet, compliant with the standards described in the IEEE 802.3af specification for delivering in-line power. The 802.3af specification defines the standard for delivering power over existing network cabling infrastructure, enabling multicast-enabled full streaming audio and video applications for converged services, such as, Voice over IP (VoIP), WLAN access points, IP surveillance cameras, and other IP technology devices.

POE technology eliminates the need for an electrical outlet and dedicated UPS near IP powered devices. With power sourcing devices, such as Foundry's FSX, power is consolidated and centralized in the wiring closets, improving the reliability and resiliency of the network. Because POE can provide power over Ethernet cable, power is continuous, even in the event of a power failure.

Terms Used in This Section

The following terms are introduced in this section:

- **Power sourcing device/equipment** - This is the source of the power, or the device that integrates the power onto the network. Power sourcing devices/equipment have embedded POE technology. In this case, the power sourcing device is Foundry's POE device.
- **IP powered device** or **Power consuming device** - This is the Ethernet device that requires power and is situated on the other end of the cable opposite the power sourcing equipment.

Methods for Delivering POE

There are two methods for delivering power over the network, as defined in the 802.3af specification:

- **Endspan** - Power is supplied through the Ethernet ports on a power sourcing device. With the Endspan solution, power can be carried over the two data pairs (Alternative A) or the two spare pairs (Alternative B).
- **Midspan** - Power is supplied by an intermediate power sourcing device placed between the switch and the powered device. With the Midspan solution, power is carried over the two spare pairs (Alternative B).

With both methods, power is transferred over four conductors, between the two pairs. 802.3af-compliant powered devices are able to accept power from either pairs.

Foundry's POE devices use the Endspan method, compliant with the 802.3af standard.

The Endspan and Midspan methods are described in more detail in the following sections.

NOTE: All 802.3af-compliant power consuming devices are required to support both application methods defined in the 802.3af specification.

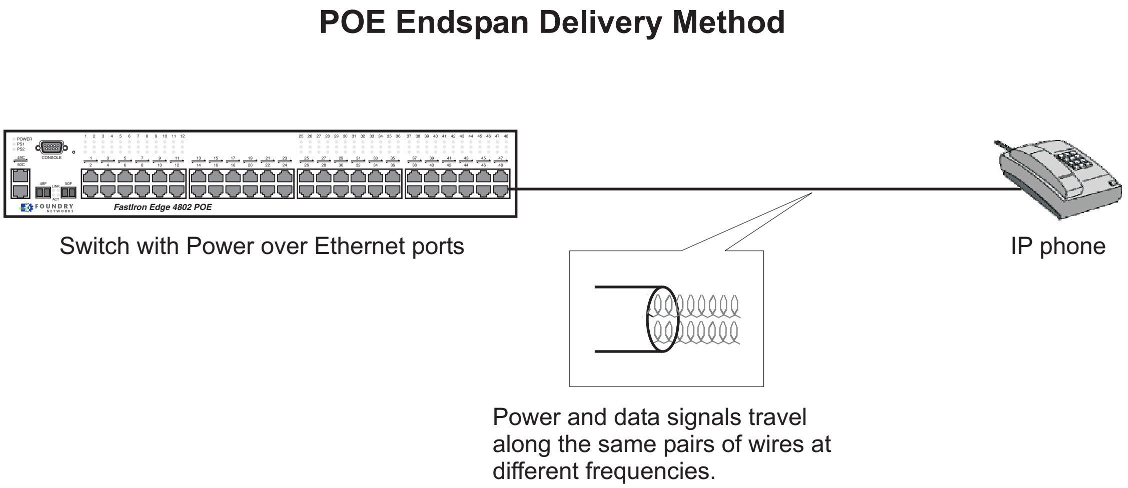
Endspan

The POE Endspan method uses the Ethernet switch ports on power sourcing equipment, such as Foundry's FSX POE, which has embedded POE technology to deliver power over the network.

With the Endspan solution, there are two supported methods of delivering power. In Alternative A, four wires deliver data and power over the network. Specifically, power is carried over the live wire pairs that deliver data, as illustrated in Figure 11.1. In Alternative B, the four wires of the spare pairs are used to deliver power over the network. Foundry's POE devices support Alternative A.

The Endspan method is illustrated in Figure 11.1.

Figure 11.1 POE Endspan Delivery Method

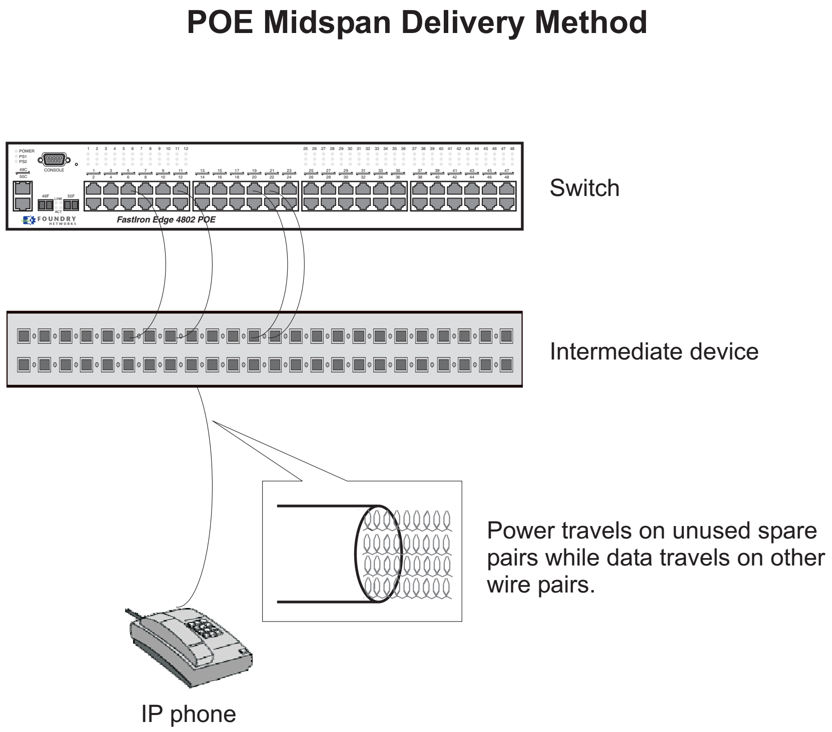

POE Endspan Delivery Method
FastIron Edge 4802 POE
FOUNDRY NETWORKS
Switch with Power over Ethernet ports
IP phone
Power and data signals travel along the same pairs of wires at different frequencies.

Midspan

The POE Midspan solution uses an intermediate device, usually a powered device, to inject power into the network. The intermediate device is positioned between the switch and the powered device and delivers power over the network using the spare pairs of wires (Alternative B). The intermediate device has multiple channels (typically 6 to 24), and each of the channels has data input and a data plus power RJ-45 output connector.

The Midspan method is illustrated in Figure 11.2.

Figure 11.2 POE Midspan Delivery Method


POE Midspan Delivery Method
FOUNDRY NETWORKS
FastIron Edge 4802 POE
Switch
Intermediate device
Power travels on unused spare pairs while data travels on other wire pairs.
IP phone

Autodiscovery

POE autodiscovery is a detection mechanism that identifies whether or not an installed device is 802.3af compatible. When you plug a device into an Ethernet port that is capable of providing in-line power, the autodiscovery mechanism detects whether or not the device requires power and how much power is needed. The autodiscovery mechanism also has a disconnect protection mechanism that shuts down the power once a powered device has been disconnected from the network or when a faulty powered device has been detected. This feature enables safe installation and prevents high-voltage damage to equipment.

POE autodiscovery is achieved by periodically transmitting current or test voltages that can detect when a powered device is attached to the network. When an 802.3af compatible device is plugged into a POE port, the powered device reflects test voltage back to the power sourcing device (the Foundry device), ultimately causing the power to be switched ON. Non-compatible 802.3af devices do not reflect test voltage back to the power sourcing device.

Power Class

Different power classes determine the amount of power a POE powered device receives. When a valid powered device is detected, the Foundry POE device performs power classification by inducing a specific voltage and measuring the current consumption of the powered device. Depending on the measured current, the Foundry device assigns the appropriate class to the powered device. Powered devices that do not support classification are assigned a class of 0 (zero). Table 11.2 shows the different power classes and their respective power consumption needs.

Table 11.1: Power Classes for Powered Devices

Class	Usage	Power (Watts)
0	default	15.4

Table 11.1: Power Classes for Powered Devices (Continued)

Class	Usage	Power (Watts)
1	optional	4
2	optional	7
3	optional	15.4
4	future	class 0

Power Specifications

The actual implementation of the 802.3af standard limits power to 15.4W (44V to 57V) from the power sourcing device. This is in compliance with safety standards and existing wiring limitations. Though limited by the 802.3af standard, 15.4 watts of power is ample, as most powered devices consume an average of 5 to 12 watts of power. IP phones, wireless LAN access points, and network surveillance cameras each consume an average of 3.5 to 9 watts of power.

The FSX's 48-volt power supply (part number SX-POE-AC-PWR) provides power to the POE daughter card, and ultimately to POE power-consuming devices. The number of POE power-consuming devices that one 48-volt power supply can support depends on the number of watts required by each power-consuming device. Each 48-volt power supply can provide 1080 watts of power, and each POE port supports a maximum of 15.4 watts of power per POE power-consuming device. For example, if each POE power-consuming device attached to the FSX consumes 10 watts of power, one 48-volt supply will power up to 108 POE ports. You can install a second 48-volt supply for additional POE power. Power supply specifications are covered in the *Foundry FastIron X Series Chassis Hardware Installation Guide* and in the *Foundry FastIron Stackable Hardware Installation Guide*.

CAUTION: The SX-POE-AC-PWR power supply is designed exclusively for use with the FSX POE devices. The power supply produces extensive power to support 802.3af applications. Installing the power supply in a device other than the FSX POE will cause extensive damage to your equipment.

Cabling Requirements

The 802.3af standard currently supports POE on 10/100/1000 Mbps Ethernet ports operating over standard Category 5 unshielded twisted pair (UTP) cable or better. If your network uses cabling categories less than 5, you cannot implement POE without first upgrading your cables to CAT 5 UTP or better.

Supported Powered Devices

Foundry's POE devices support the following types of IP powered devices:

- Voice over IP (VoIP) phones
- Wireless LAN access points
- IP surveillance cameras

The following sections briefly describe these IP powered devices.

VoIP

Voice over IP (VoIP) is the convergence of traditional telephony networks with data networks, utilizing the existing data network infrastructure as the transport system for both services. Traditionally, voice is transported on a network that uses circuit-switching technology, whereas data networks are built on packet-switching technology. To achieve this convergence, technology has been developed to take a voice signal, which originates as an analog signal and transport it within a digital medium. This is done by devices, such as VoIP Telephones, which receive the originating tones and place them in UDP packets, the size and frequency of which is dependant on the Coding / Decoding (CODEC) technology that has been implemented in the VoIP Telephone / device. The VoIP control packets use the TCP/IP format.

Wireless LAN Access Points

Wireless LANs enable you to establish and maintain a wireless network connection within or between buildings, without the constraints of wires or cables as imposed by a wired LAN. Wireless LAN access points provide the link between the wired LAN and the wireless LAN.

Foundry's IronPoint™ Access Point allows wireless clients to connect to your enterprise network. It is a full-featured access point that can be managed as a single device or by IronView Network Manager, a network management tool that manages several Foundry devices on a network. For more information about Foundry's IronPoint Access Point, see the *Foundry IronPoint Wireless LAN Configuration Guide*.

One of the main concerns with wireless LAN access points is the additional protection needed to secure the network. To help ensure continuous security against unauthorized Wireless LAN Access Points deployment, and deliver advanced security for entry-level WLAN Access Points, the Foundry's POE devices include IEEE 802.1x support for a flexible and dynamic security implementation. All switch ports can be configured as secured, requiring 802.1x authentication, or unsecured, requiring no authentication. For more information about this feature, refer to the chapter "Configuring 802.1X Port Security" on page 42-1.

IP Surveillance Cameras

IP surveillance technology provides digital streaming of video over Ethernet, providing real-time, remote access to video feeds from cameras.

The main benefit of using IP surveillance cameras on the network is that you can view surveillance images from any computer on the network. If you have access to the Internet, you can securely connect from anywhere in the world to view a chosen facility or even a single camera from your surveillance system. By using a Virtual Private Network (VPN) or the company intranet, you can manage password-protected access to images from the surveillance system. Similar to secure payment over the Internet, images and information are kept secure and can be viewed only by approved personnel.

Enabling or Disabling Power over Ethernet

To enable a port to receive in-line power for 802.3af-compliant and non-compliant power consuming devices, enter commands such as the following:

```
FastIron#config t
FastIron(config)#interface e 1/1
FastIron(config-if-e1000-1/1)#inline power
```

After entering the above commands, the console will display the following message:

```
FastIron(config-if-e1000-1/1)#PoE Info: Power enabled on port 1/1.
```

Syntax: [no] inline power

Use the **no** form of the command to disable the port from receiving in-line power.

NOTE: The FSX with POE can automatically detect whether or not a power consuming device is 802.3af-compliant. Therefore, the CLI command **inline power legacy-powerdevice**, which is used on FES POE devices to configure 802.3af non-compliant devices, does not apply on the FSX POE.

Disabling Support for POE Legacy Power Consuming Devices

This feature is supported on FastIron X Series POE devices running software release 03.0.00 or later.

Foundry's POE devices automatically support most legacy power consuming devices (non-802.3af compliant devices), as well as all 802.3af-compliant devices. In releases prior to 03.0.00, support for legacy devices is always enabled on Foundry's POE devices and cannot be disabled. Starting with release 03.0.00, if desired, you can disable and re-enable support for legacy POE power consuming devices on a global basis (on the entire device) or on individual slots (chassis devices only). When you disable legacy support, 802.3af-compliant devices are not affected.

To disable support for legacy power consuming devices on a global basis, enter the following command at the Global CONFIG level of the CLI:

```
FastIron(config)#no legacy-inline-power
```

On chassis devices (FSX, FSX 800, and FSX 1600), you can disable support for legacy power consuming devices per slot. To disable support on a slot, enter a command such as the following at the Global CONFIG level of the CLI:

```
FastIron(config)#no legacy-inline-power 2
```

The above command disables legacy support on all ports in slot 2.

Syntax: [no] legacy-inline-power [<slotnum>/]

To re-enable support for legacy power consuming devices after it has been disabled, enter the **legacy-inline-power** command (without the **no** parameter).

<slotnum> is required on chassis devices when disabling or re-enabling legacy support on a slot.

Use the **show run** command to view whether support for POE legacy power consuming devices is enabled or disabled.

Enabling the Detection of POE Power Requirements Advertised via CDP

Many power consuming devices, such as Cisco's VOIP phones and other vendors' devices, use CDP to advertise their power requirements to power sourcing devices, such as Foundry's POE devices. Foundry's power consuming devices are compatible with Cisco's and other vendors' power consuming devices, in that they can detect and process power requirements for these devices automatically.

Configuration Considerations

- This feature is supported in FSX POE devices running software release 02.2.00 or later
- If you configure a port with a maximum power level or a power class for a power consuming device, the power level or power class takes precedence over the CDP power requirement. Therefore, if you want the device to adhere to the CDP power requirement, do not configure a power level or power class on the port.
- The FSX POE will adjust a port's power only if there are available power resources on the device.

Command Syntax

To enable the Foundry device to detect CDP power requirements, enter the following commands:

```
FastIron#config t
FastIron(config)#cdp run
```

Syntax: [no] cdp run

Use the **no** form of the command to disable the detection of CDP power requirements.

Setting the Maximum Power Level for a POE Power Consuming Device

When POE is enabled on a port to which a power consuming device is attached, by default, the Foundry POE device will supply 15.4 watts of power at the RJ45 jack, minus any power loss through the cables. For example, a POE port with a default maximum power level of 15.4 watts will receive a maximum of 12.95 watts of power after 2.45 watts of power loss through the cable. This is compliant with the IEEE 802.3af specification for delivering in-line power. Devices that are configured to receive less POE power, for example, 4.0 watts of power, will experience a lower rate of power loss through the cable.

If desired, you can manually configure the maximum amount of power that the Foundry POE device will supply at the RJ45 jack. You can specify from 1 to 15.4 watts of maximum power for each power consuming device connected to the switch.

Configuration Notes

- This feature is supported in FSX POE devices running release 02.2.00 or later
- There are two ways to configure the power level for a POE power consuming device. The first method is discussed in this section. The other method is provided in the section “Setting the Power Class for a POE Power Consuming Device” on page 11-7. For each POE port, you can configure either a maximum power level or a power class. You cannot configure both. You can, however, configure a maximum power level on one port and a power class on another port.
- The CLI commands for this feature differ on the FSX POE compared to the FES POE. On the FES POE, there are separate CLI commands for 802.3af-compliant versus 802.3-af non-compliant power consuming devices. On the FSX, there is one command for all power consuming devices. The command syntax is also different on the FSX. To configure your device, refer to the appropriate section, below.

Command Syntax

To configure the maximum power level for a power consuming device, enter commands such as the following:

```
FastIron#config t
FastIron(config)#interface e 1/1
FastIron(config-if-e1000-1/1)#inline power power-limit 14000
```

These commands enable in-line power on interface e 1 in slot 1 and set the POE power level to 14,000 milliwatts (14 watts).

Syntax: inline power power-limit <power level>

where <power level> is the number of milliwatts, between 1000 and 15400. The default is 15400.

For information about resetting the maximum power level, see “Resetting POE Parameters” on page 11-9.

Setting the Power Class for a POE Power Consuming Device

A power class specifies the maximum amount of power that a Foundry POE device will supply to a power consuming device. Table 11.2 shows the different power classes and their respective maximum power allocations.

Table 11.2: Power Classes for Power Consuming Devices

Class	Maximum Power (Watts)
0	15.4 (default)
1	4
2	7
3	15.4

By default, the power class for all power consuming devices is zero (0). As shown in Table 11.2, a power consuming device with a class of 0 receives 15.4 watts of power.

Configuration Notes

- This feature is supported in the FSX POE devices running release 02.2.00 or later

- The power class sets the maximum power level for a power consuming device. Alternatively, you can set the maximum power level as instructed in the section “Setting the Maximum Power Level for a POE Power Consuming Device” on page 11-6. For each POE port, you can configure either a power class or a maximum power level. You cannot configure both. You can, however, configure a power level on one port and power class on another port.
- The power class includes any power loss through the cables. For example, a POE port with a default power class of 0 (15.4 watts) will receive a maximum of 12.95 watts of power after 2.45 watts of power loss through the cable. This is compliant with the IEEE 802.3af specification for delivering in-line power. Devices that are configured to receive less POE power, for example, class 1 devices (4.0 watts), will experience a lower rate of power loss through the cable.
- The CLI commands for this feature differ on the FSX POE compared to the FES POE. On the FES POE, there are separate CLI commands for 802.3af-compliant versus 802.3-af non-compliant power consuming devices. On the FSX, there is one command for all power consuming devices. The command syntax is also different on the FSX.

Command Syntax

To configure the power class for a POE power consuming device, enter commands such as the following:

```
FastIron#config t
FastIron(config)#interface e 1/1
FastIron(config-if-e1000-1/1)#inline power power-by-class 2
```

These commands enable in-line power on interface e 1 in slot 1 and set the power class to 2.

Syntax: inline power power-by-class <class value>

where <class value> is the power class. Enter a value from 0 – 3. See Table 11.2 for the power classes and their respective maximum power allocations. The default is 0 (15.4 watts).

For information about resetting the power class, see “Resetting POE Parameters” on page 11-9.

Setting the In-line Power Priority for a POE Port

Each FSX POE (48V) power supply provides a maximum of 1080 watts of power, and each POE port receives a default maximum value of 15.4 watts of power, minus any power loss through the cable. The power capacity of one or two POE power supplies is shared among all POE power consuming devices attached to the FSX POE.

In a configuration where POE power consuming devices collectively have a greater demand for power than the POE power supply or supplies can provide, the FSX must place the POE ports that it cannot power in *standby* or *denied* mode (waiting for power) until the available power increases. The available power increases when one or more POE ports are powered down, or, if applicable, when an additional POE power supply is installed in the FSX.

When POE ports are in *standby* or *denied* mode (waiting for power) and the FSX receives additional power resources, by default, the FSX will allocate newly available power to the standby ports in ascending order, by slot number then by port number, provided enough power is available for the ports. For example, POE port 1/11 should receive power before POE port 2/1. However, if POE port 1/11 needs 12 watts of power and POE port 2/1 needs 10 watts of power, and 11 watts of power become available on the device, the FSX will allocate the power to port 2/1 since it does not have sufficient power for port 1/11.

You can configure an *in-line power priority* on POE ports, whereby ports with a higher in-line power priority will take precedence over ports with a low in-line power priority. For example, if a new POE port comes on-line and the port is configured with a high priority, if necessary (if power is already fully allocated to power consuming devices), the FSX will remove power from a POE port or ports that have a lower priority and allocate the power to the POE port that has the higher value.

Ports that are configured with the same in-line power priority are given precedence based on the slot number and port number in ascending order, provided enough power is available for the port. For example, if both POE port 1/2 and POE port 2/1 have a high in-line power priority value, POE port 1/2 will receive power before POE port 2/1.

However, if POE port 1/2 needs 12 watts of power and POE port 2/1 needs 10 watts of power, and 11 watts of power become available on the device, the FSX will allocate the power to POE port 2/1 since it does not have sufficient power for port 1/2. By default, all ports are configured with a low in-line power priority.

Command Syntax

To configure an in-line power priority for a POE port on a FSX, enter commands such as the following:

```
FastIron#config t
FastIron(config)#interface e 1/1
FastIron(config-if-e1000-1/1)#inline power priority 2
```

These commands enable in-line power on interface e 1 in slot 1 and set the in-line power priority level to high.

Syntax: [no] inline power priority <priority num>

where **priority <priority num>** is the in-line power priority number. The default is 3 (low priority). You can specify one of the following values:

- 3 – low priority
- 2 – high priority
- 1 – critical priority

Use the **inline power** command (without a priority number) to reset a port's priority to the default (low) priority.

Use the **no inline power** command to disable the port from receiving in-line power.

For information about resetting the in-line power priority, see "Resetting POE Parameters" on page 11-9.

To view the in-line power priority for all POE ports, issue the **show inline power** command at the Privileged EXEC level of the CLI. See "Displaying POE Operational Status" on page 11-10.

Resetting POE Parameters

NOTE: This feature applies to the FSX POE only.

To override or reset POE port parameters including power priority, power class, and maximum power level, you must specify each POE parameter in the CLI command line. This section provides some examples.

EXAMPLES:

To change a POE port's power priority from high to low (the default value) and keep the current maximum configured power level of 3000, enter commands such as the following:

```
FastIron#config t
FastIron(config)#interface e 1/1
FastIron(config-if-e1000-1/1)#inline power priority 3 power-limit 3000
```

Note that you must specify both the inline power priority and the maximum power level (**power-limit** command), even though you are keeping the current configured maximum power level at 3000. If you do not specify the maximum power level, the device will apply the default value of15400 (15.4 watts). Also, you must specify the inline power priority before specifying the power limit.

EXAMPLES:

To change a port's power class from 2 (4 watts max) to 3 (7 watts max) and keep the current configured power priority of 2, enter commands such as the following:

```
FastIron#config t
FastIron(config)#interface e 1/1
FastIron(config-if-e1000-1/1)#inline power priority 2 power-by-class 3
```

Note that you must specify both the power class and the inline power priority, even though you are not changing the power priority. If you do not specify the power priority, the device will apply the default value of 3 (low priority). Also, you must specify the inline power priority before specifying the power class.

Displaying Power over Ethernet Information

This section lists the CLI commands for viewing POE information.

Displaying POE Operational Status

The **show inline power** command displays operational information about Power over Ethernet.

On the FSX, you can view the POE operational status for the entire device, for a specific POE module only, or for a specific interface only. In addition, on the FSX, you can use the **show inline power detail** command to display in depth information about POE power supplies.

The following shows an example of the **show inline power** display output on a FSX device.

```
FastIron#show inline power

Power Capacity:         Total is 2160000 mWatts. Current Free is 18800 mWatts.

Power Allocations:      Requests Honored 769 times

 ... some lines omitted for brevity...

 Port   Admin   Oper    ---Power(mWatts)---  PD Type  PD Class  Pri  Fault/
        State   State   Consumed  Allocated                          Error
--------------------------------------------------------------------------
 4/1    On      On         5070       9500  802.3af  n/a         3  n/a
 4/2    On      On         1784       9500  Legacy   n/a         3  n/a
 4/3    On      On         2347       9500  802.3af  n/a         3  n/a
 4/4    On      On         2441       9500  Legacy   n/a         3  n/a
 4/5    On      On         6667       9500  802.3af  Class 3     3  n/a
 4/6    On      On         2723       9500  802.3af  Class 2     3  n/a
 4/7    On      On         2347       9500  802.3af  n/a         3  n/a
 4/8    On      On         2347       9500  802.3af  n/a         3  n/a
 4/9    On      On         2347       9500  802.3af  n/a         3  n/a
 4/10   On      On         4976       9500  802.3af  Class 3     3  n/a
 4/11   On      On         4882       9500  802.3af  Class 3     3  n/a
 4/12   On      On         4413       9500  802.3af  Class 1     3  n/a
 4/13   On      On         7793       9500  802.3af  n/a         3  n/a
 4/14   On      On         7512       9500  802.3af  n/a         3  n/a
 4/15   On      On         8075       9500  802.3af  n/a         3  n/a
 4/16   On      On         4131       9500  802.3af  Class 1     3  n/a
 4/17   On      On         2347       9500  802.3af  n/a         3  n/a
 4/18   On      Off           0       9500  n/a      n/a         3  n/a
 4/19   On      On         5352       9500  Legacy   n/a         3  n/a
 4/20   On      On         7981       9500  802.3af  n/a         3  n/a
 4/21   On      On        12958      13000  802.3af  Class 3     3  n/a
 4/22   On      On        12958      13000  802.3af  Class 3     3  n/a
 4/23   On      On        13052      13000  802.3af  Class 3     3  n/a
 4/24   On      On        12864      13000  802.3af  Class 3     3  n/a
--------------------------------------------------------------------------
 Total                   137367     242000

 ... some lines omitted for brevity...

 Grand Total            1846673    2127400
```

Syntax: show inline power [<slotnum>] | [<slotnum>/<portnum>]

Table 11.3 provides definitions for the statistics.

Table 11.3: Field Definitions for the Show Inline Power Command

This Column...	Displays...
Power Capacity	The total POE power supply capacity and the amount of available power (current free) for POE power consuming devices. Both values are shown in milliwatts.
Power Allocations	The number of times the FSX fulfilled POE requests for power.

Table 11.3: Field Definitions for the Show Inline Power Command

This Column...	Displays...
Port	The slot number and port number.
Admin State	Specifies whether or not Power over Ethernet has been enabled on the port. This value can be one of the following: • ON – The **inline power** command was issued on the port. • OFF – The **inline power** command has not been issued on the port.
Oper State	Shows the status of in-line power on the port. This value can be one of the following: • ON – The POE power supply is delivering in-line power to the powered device. • OFF – The POE power supply is not delivering in-line power to the powered device. • DENIED – The port is in standby mode (waiting for power) because the FSX does not currently have enough available power for the port. **NOTE:** In FSX software release 03.2.00 and later, when you enable a port using the CLI, it may take 12 or more seconds before the operational state of that port is displayed correctly in a **show inline power** output.
Power Consumed	The number of current, actual milliwatts that the powered device is consuming.
Power Allocated	The number of milliwatts allocated to the port. This value is either the default or configured maximum power level, or the power class that was automatically detected by the FSX.
PD Type	The type of powered device connected to the port. This value can be one of the following: • 802.3AF-PD – The powered device connected to this port is 802.3af-compliant. • LEGACY – The powered device connected to this port is a legacy product (not 802.3af-compliant). • N/A – Power over Ethernet is configured on this port, and one of the following is true: • The device connected to this port is a non-powered device. • No device is connected to this port. • The port is in *standby* or *denied* mode (waiting for power).
PD Class	Determines the maximum amount of power a powered device receives. This value can be one of the following: • Class0 – Receives 15.4 watts maximum. • Class1 – Receives 4 watts maximum • Class2 – Receives 7 watts maximum • Class3 – Receives 15.4 watts maximum • Unknown – The device attached to the port cannot advertise its class.

Table 11.3: Field Definitions for the Show Inline Power Command

This Column...	Displays...
Pri	The port's *in-line power priority*, which determines the order in which the port will receive power while in standby mode (waiting for power). Ports with a higher priority will receive power before ports with a low priority. This value can be one of the following: • 3 – low priority • 2 – high priority • 1 – critical priority
Fault/Error	If applicable, this is the fault or error that occurred on the port. This value can be one of the following: • critical temperature – The POE chip temperature limit rose above the safe operating level, thereby powering down the port. • detection failed - discharged capacitor – The port failed capacitor detection (legacy PD detection) because of a discharged capacitor. This can occur when connecting a non-PD on the port. • detection failed - out of range capacitor – The port failed capacitor detection (legacy PD detection) because of an out-of-range capacitor value. This can occur when connecting a non-PD on the port. • internal h/w fault – A hardware problem has hindered port operation. • lack of power – The port has shut down due to lack of power. • main supply voltage high – The voltage was higher than the maximum voltage limit, thereby tripping the port. • main supply voltage low – The voltage was lower than the minimum voltage limit, thereby tripping the port. • overload state – The PD consumes more power than the maximum limit configured on the port, based on the default configuration, user configuration, or CDP configuration. • over temperature – The port temperature rose above the temperature limit, thereby powering down the port. • PD DC fault – A succession of underload and overload states, or a PD's DC/DC fault, caused the port to shutdown. • short circuit – A short circuit was detected on the port delivering power. • underload state – The PD consumes less power than the minimum limit specified in the 802.3af standard. • voltage applied from ext src – The port failed capacitor detection (legacy PD detection) because the voltage applied to the port was from an external source.
Total	The total power in milliwatts being *consumed* by all powered devices connected to the Interface module, and the total power in milliwatts *allocated* to all powered devices connected to the Interface module.
Grand Total	The total number of current, actual milliwatts being *consumed* by all powered devices connected to the FSX, and the total number of milliwatts *allocated* to all powered devices connected to the FSX.

Displaying Detailed Information about POE Power Supplies

The **show inline power detail** command displays detailed operational information about the POE power supplies in **FSX** POE devices.

To display detailed POE statistics, enter the following command:

```
FastIron#show inline power detail

Power Supply Data:
++++++++++++++++++

Power Supply #1:
        Firmware Ver:   0.2
        Date:           3/15/5
        H/W Status:     807
        Max Curr:       26.5 Amps
        Voltage:        50.0 Volts
        Capacity:       1325 Watts
        Consumption:    1144 Watts
Power Supply #2:
        Firmware Ver:   0.2
        Date:           3/15/5
        H/W Status:     807
        Max Curr:       26.5 Amps
        Voltage:        50.0 Volts
        Capacity:       1325 Watts
        Consumption:    949 Watts

General PoE Data:
+++++++++++++++++

Slot  Firmware
      Version
--------------
1     04.0.0
2     04.0.0
3     04.0.0
4     04.0.0
5     04.0.0
6     04.0.0
7     04.0.0
8     04.0.0

... continued on next page...
... continued from previous page...

Cumulative Port State Data:
+++++++++++++++++++++++++++

Slot  #Ports    #Ports     #Ports   #Ports    #Ports       #Ports     #Ports
      Admin-On  Admin-Off  Oper-On  Oper-Off  Off-Denied   Off-No-PD  Off-Fault
--------------------------------------------------------------------------------
1     24        0          24       0         0            0          0
2     24        0          24       0         0            0          0
3     24        0          23       1         0            1          0
4     24        0          23       1         0            1          1
5     24        0          24       0         0            0          0
6     24        0          24       0         0            0          0
7     24        0          24       0         0            0          0
8     24        0          24       0         0            0          0
--------------------------------------------------------------------------------
Total:192       0          190      2         0            2          1


Cumulative Port Power Data:
+++++++++++++++++++++++++++

Slot  #Ports  #Ports  #Ports        Power       Power
      Pri: 1  Pri: 2  Pri: 3  Consumption  Allocation
-----------------------------------------------------
1     24      0       0         310.146 W   312.0   W
2     0       0       24        308.454 W   312.0   W
3     0       0       24        108.727 W   172.500 W
4     0       0       24        137.366 W   232.500 W
5     24      0       0          56.991 W   145.400 W
6     0       0       24        309.112 W   312.0   W
7     0       0       24        308.548 W   312.0   W
8     24      0       0         307.796 W   312.0   W
-----------------------------------------------------
Total:72      0       120      1847.140 W  2110.400 W
```

Syntax: show inline power detail

Table 11.3 provides definitions for the statistics displayed in the **show inline power detail** command.

Table 11.4: Field Definitions for the Show Inline Power Detail Command

This Column...	Displays...
Power Supply Data	
Firmware ver	The POE power supply's firmware version.
Date	The POE power supply's firmware test date in the format mm/dd/yyyy.
H/W Status	The POE power supply's hardware status code. This field is used by Foundry Technical Support for troubleshooting.
Max Curr	The POE power supply's maximum current capacity.

Table 11.4: Field Definitions for the Show Inline Power Detail Command (Continued)

This Column...	Displays...
Voltage	The POE power supply’s current input voltage.
Capacity	The POE power supply’s total power capacity (in watts).
Consumption	The total number of watts consumed by POE power consuming devices and POE modules in the system, minus any internal or cable power loss.
General POE Data	
Slot	The Interface module / slot number
Firmware Version	The Interface module’s / slot number’s firmware version.
Cumulative Port State Data **NOTE:** In FSX software release 03.2.00 and later, when you enable a port using the CLI, it may take 12 or more seconds before the operational state of that port is displayed correctly in a **show inline power** output.	
Slot	The Interface module / slot number
#Ports Admin-On	The number of ports on the Interface module on which the **inline power** command was issued.
#Ports Admin-Off	The number of ports on the Interface module on which the **inline power** command was not issued.
#Ports Oper-On	The number of ports on the Interface module that are receiving in-line power from the POE power supply.
#Ports Oper-Off	The number of ports on the Interface module that are not receiving in-line power from the POE power supply.
#Ports Off-Denied	The number of ports on the Interface module that were denied power because of insufficient power.
#Ports Off-No-PD	The number of ports on the Interface module to which no powered devices are connected.
#Ports Off-Fault	The number of ports on the Interface module that are not receiving power because of a subscription overload.
Total	The totals for all of the fields in the **Cumulative Port State Data** report.
Cumulative Port Power Data	
Slot	**The Interface module / slot number**
#Ports Pri: 1	The number of POE ports on the Interface module that have a POE port priority of 1.
#Ports Pri: 2	The number of POE ports on the Interface module that have a POE port priority of 2.
#Ports Pri: 3	The number of POE ports on the Interface module that have a POE port priority of 3.
Power Consumption	The total number of watts consumed by both POE power consuming devices and the POE module (daughter card) attached to the Interface module.

Table 11.4: Field Definitions for the Show Inline Power Detail Command (Continued)

This Column...	Displays...
Power Allocation	The number of watts allocated to the Interface module’s POE ports. This value is the sum of the ports’ default or configured maximum power levels, or power classes automatically detected by the FSX.
Total	The totals for all of the fields in the **Cumulative Port Power Data** report.

Chapter 12

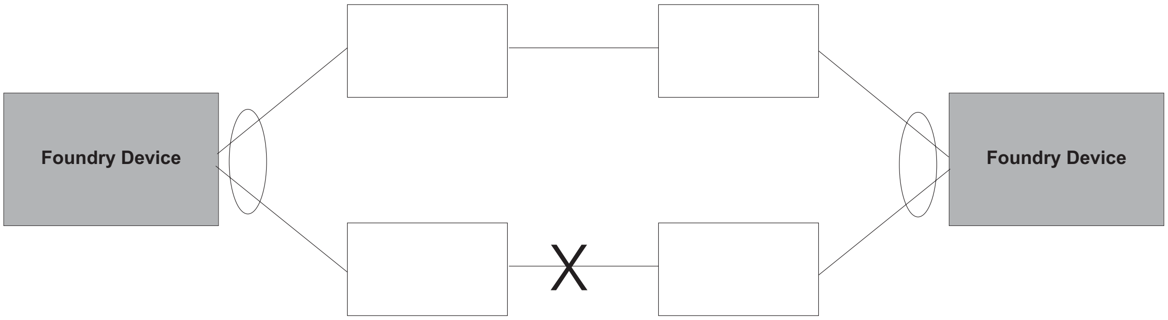
Configuring Uni-Directional Link Detection (UDLD) and Protected Link Groups

This chapter describes how to configure Uni-Directional Link Detection (UDLD) and Protected Link Groups on a Foundry FastIron X Series device using the CLI.

UDLD Overview

Uni-Directional Link Detection (UDLD) monitors a link between two Foundry devices and brings the ports on both ends of the link down if the link goes down at any point between the two devices. This feature is useful for links that are individual ports and for trunk links. Figure 12.1 shows an example.

Figure 12.1 UDLD Example



Without link keepalive, the Foundry ports remain enabled. Traffic continues to be load balanced to the ports connected to the failed link.

When link keepalive is enabled, the feature brings down the Foundry ports connected to the failed link.

Normally, a Foundry device load balances traffic across the ports in a trunk group. In this example, each Foundry device load balances traffic across two ports. Without the UDLD feature, a link failure on a link that is not directly attached to one of the Foundry devices is undetected by the Foundry devices. As a result, the Foundry devices continue to send traffic on the ports connected to the failed link.

When UDLD is enabled on the trunk ports on each Foundry device, the devices detect the failed link, disable the ports connected to the failed link, and use the remaining ports in the trunk group to forward the traffic.

Ports enabled for UDLD exchange proprietary health-check packets once every second (the keepalive interval). If a port does not receive a health-check packet from the port at the other end of the link within the keepalive interval, the port waits for two more intervals. If the port still does not receive a health-check packet after waiting for three intervals, the port concludes that the link has failed and takes the port down.

Configuration Considerations

- This feature is supported only on Ethernet ports.
- To configure UDLD on a trunk group, you must enable and configure the feature on each port of the group individually. Configuring UDLD on a trunk group's primary port enables the feature on that port only.
- Dynamic trunking is not supported. If you want to configure a trunk group that contains ports on which UDLD is enabled, you must remove the UDLD configuration from the ports. After you create the trunk group, you can re-add the UDLD configuration.
- If MRP is also enabled on the device, Foundry recommends that you set the MRP preforwarding time slightly higher than the default of 300 ms; for example, to 400 or 500 ms. See “Changing the Hello and PreForwarding Times” on page 10-15.

Enabling UDLD

To enable UDLD on a port, enter a command such as the following at the global CONFIG level of the CLI:

```
FastIron(config)#link-keepalive ethernet 0/1/1
```

To enable the feature on a trunk group, enter commands such as the following:

```
FastIron(config)#link-keepalive ethernet 0/1/1 ethernet 0/1/2
FastIron(config)#link-keepalive ethernet 0/1/3 ethernet 0/1/4
```

Syntax: [no] link-keepalive ethernet [slotnum/portnum] ethernet [slotnum/portnum] (The slotnum parameter is required on chassis devices).

These commands enable UDLD on ports 1/1 – 1/4.

For releases prior to FGS 03.0.00, you can specify up to two ports on the same command line. FGS software release 03.0.00 adds the capability to configure multiple ports with a single command, such as the following:

```
FastIron(config)#link-keepalive ethernet 0/1/1 to 0/1/5 ethernet 0/1/7
```

Syntax: [no] link-keepalive ethernet [portlist]

Changing the Keepalive Interval

By default, ports enabled for UDLD send a link health-check packet once every 500 ms. You can change the interval to a value from 1 – 60, where 1 is 100 ms, 2 is 200 ms, and so on. To change the interval, enter a command such as the following:

```
FastIron(config)#link-keepalive interval 3
```

Syntax: [no] link-keepalive interval <num>

The <num> parameter specifies how often the ports send a UDLD packet. You can specify from 1 – 60, in 100 ms increments. The default is 5 (500 ms).

Changing the Keepalive Retries

By default, a port waits one second to receive a health-check reply packet from the port at the other end of the link. If the port does not receive a reply, the port tries four more times by sending up to four more health-check packets. If the port still does not receive a reply after the maximum number of retries, the port goes down.

You can change the maximum number of keepalive attempts to a value from 3 – 10. To change the maximum number of attempts, enter a command such as the following:

```
FastIron(config)#link-keepalive retries 4
```

Syntax: [no] link-keepalive retries <num>

The <num> parameter specifies the maximum number of times the port will try the health check. You can specify a value from 3 – 10. The default is 5.

UDLD for Tagged Ports

Platform Support:

- FESX/FSX/FWSX devices running software release 04.0.00 and later – L2, BL3, L3

The default implementation of UDLD sends the packets untagged, even across tagged ports. If the untagged UDLD packet is received by a third-party switch, that switch may reject the packet. As a result, UDLD may be limited only to Foundry devices, since UDLD may not function on third-party switches.

You can configure ports to send out UDLD control packets that are tagged with a specific VLAN ID as tagged UDLD control packets. This feature also enables third party switches to receive the control packets that are tagged with the specified VLAN.

To enable ports to receive and send UDLD control packets tagged with a specific VLAN ID, enter commands such as the following:

```
FastIron(config)#link-keepalive ethernet 1/18 vlan 22
```

This command enables UDLD on port 1/18 and allows UDLD control packet tagged with VLAN 22 to be received and sent on port 1/18.

Syntax: [no] link-keepalive ethernet [<slotnum>/]<portnum> [vlan <vlan-ID>]

The <slotnum> parameter is required on chassis devices.

Enter the ID of the VLAN that the UDLD control packets can contain to be received and sent on the port. If a VLAN ID is not specified, then UDLD control packets are sent out of the port as untagged packets.

NOTE: You must configure the same VLANs that will be used for UDLD on all devices across the network; otherwise, the UDLD link cannot be maintained.

Displaying UDLD Information

This section describes the commands used to display information about a UDLD configuration.

Displaying Information for All Ports

To display UDLD information for all ports, enter the following command:

```
FastIron#show link-keepalive
Total link-keepalive enabled ports: 4
Keepalive Retries: 3    Keepalive Interval: 1 Sec.

Port    Physical Link   Logical Link    State           Link-vlan
4/1     up              up              FORWARDING      3
4/2     up              up              FORWARDING
4/3     down            down            DISABLED
4/4     up              down            DISABLED
```

Syntax: show link-keepalive

Table 12.1: CLI Display of UDLD Information

This Field...	Displays...
Total link-keepalive enabled ports	The total number of ports on which UDLD is enabled.
Keepalive Retries	The number of times a port will attempt the health check before concluding that the link is down.
Keepalive Interval	The number of seconds between health check packets.
Port	The port number.
Physical Link	The state of the physical link. This is the link between the Foundry port and the directly connected device.
Logical Link	The state of the logical link. This is the state of the link between this Foundry port and the Foundry port on the other end of the link.
State	The traffic state of the port.
Link-vlan	The ID of the tagged VLAN in the UDLD packet.

If a port is disabled by UDLD, the change also is indicated in the output of the **show interfaces brief** command. Here is an example:

```
FastIron#show interfaces brief

Port  Link State     Dupl Speed Trunk Tag Priori MAC            Name
1/1   Up   LK-DISABLENone None  None  No  level0 00e0.52a9.bb00
1/2   Down None      None None  None  No  level0 00e0.52a9.bb01
1/3   Down None      None None  None  No  level0 00e0.52a9.bb02
1/4   Down None      None None  None  No  level0 00e0.52a9.bb03
```

If the port was already down before you enabled UDLD for the port, the port’s state is listed as None.

Syntax: show interfaces brief

Displaying Information for a Single Port

To display detailed UDLD information for a specific port, enter a command such as the following:

```
FastIron#show link-keepalive ethernet 4/1

Current State   : up            Remote MAC Addr  : 00e0.52d2.5100
Local Port      : 4/1           Remote Port      : 2/1
Local System ID : e0927400      Remote System ID : e0d25100
Packets sent    : 254           Packets received : 255
Transitions     : 1             Link-vlan        : 100
Port blocking   : No            BM disabled      : No
```

Syntax: show link-keepalive [ethernet [<slotnum>/]<portnum>]

Table 12.2: CLI Display of Detailed UDLD Information

This Field...	Displays...
Current State	The state of the logical link. This is the link between this Foundry port and the Foundry port on the other end of the link.
Remote MAC Addr	The MAC address of the port or device at the remote end of the logical link.
Local Port	The port number on this Foundry device.
Remote Port	The port number on the Foundry device at the remote end of the link.
Local System ID	A unique value that identifies this Foundry device. The ID can be used by Foundry technical support for troubleshooting.
Remote System ID	A unique value that identifies the Foundry device at the remote end of the link.
Packets sent	The number of UDLD health-check packets sent on this port.
Packets received	The number of UDLD health-check packets received on this port.
Transitions	The number of times the logical link state has changed between up and down.
Port blocking	Information used by Foundry technical support for troubleshooting.
Link-vlan	The ID of the tagged VLAN in the UDLD packet.
BM disabled	Information used by Foundry technical support for troubleshooting.

The **show interface ethernet** command also displays the UDLD state for an individual port. In addition, the line protocol state listed in the first line will say “down” if UDLD has brought the port down. Here is an example:

```
FastIron#show interface ethernet 1/1
FastEthernet1/1 is down, line protocol is down, link keepalive is enabled
  Hardware is FastEthernet, address is 00e0.52a9.bbca (bia 00e0.52a9.bbca)
  Configured speed auto, actual unknown, configured duplex fdx, actual unknown
  Member of L2 VLAN ID 1, port is untagged, port state is DISABLED
  STP configured to ON, priority is level0, flow control enabled
  mirror disabled, monitor disabled
  Not member of any active trunks
  Not member of any configured trunks
  No port name
  300 second input rate: 0 bits/sec, 0 packets/sec, 0.00% utilization
  300 second output rate: 0 bits/sec, 0 packets/sec, 0.00% utilization
  0 packets input, 0 bytes, 0 no buffer
  Received 0 broadcasts, 0 multicasts, 0 unicasts
  0 input errors, 0 CRC, 0 frame, 0 ignored
  0 runts, 0 giants, DMA received 0 packets
  19 packets output, 1216 bytes, 0 underruns
  Transmitted 0 broadcasts, 19 multicasts, 0 unicasts
  0 output errors, 0 collisions, DMA transmitted 19 packets
```

In this example, the port has been brought down by UDLD. Notice that in addition to the information in the first line, the port state on the fourth line of the display is listed as DISABLED.

Clearing UDLD Statistics

To clear UDLD statistics, enter the following command:

```
FastIron#clear link-keepalive statistics
```

Syntax: clear link-keepalive statistics

This command clears the Packets sent, Packets received, and Transitions counters in the **show link keepalive ethernet** [<slotnum>/]<portnum> display.

Protected Link Groups

Platform Support:

- FESX/FSX/FWSX devices running software release 03.0.00 and later – L2, BL3, L3
- FGS and FLS devices running software release 03.0.00 and later

A protected link group minimizes disruption to the network by protecting critical links from loss of data and power. In a protected link group, one port in the group acts as the primary or active link, and the other ports act as secondary or standby links. The active link carries the traffic. If the active link goes down, one of the standby links takes over.

During normal operation, the active port in a protected link group is enabled and the standby ports are logically disabled. If the active port fails, the Foundry device immediately enables one of the standby ports, and switches traffic to the standby port. The standby port becomes the new, active port.

About Active Ports

When you create a protected link group, you can optionally specify which port in the protected link group is the active port. If you do not explicitly configure an active port, the Foundry device dynamically assigns one. A dynamic active port is the first port in the protected link group that comes up (usually the lowest numbered port in the group).

Static and dynamic active ports operate as follows:

- A static active port (an active port that you explicitly configured) pre-empts other ports in the protected link group. So, if a static active link comes back up after a failure, the Foundry device will revert to this link as the active link.
- A dynamic active port (an active port assigned by the software) is non-pre-emptive. Therefore, if a dynamic active link comes back up after a failure, the Foundry device does not revert to this link, but continues carrying traffic on the current active link.

Using UDLD with Protected Link Groups

You can use UDLD with protected link groups to detect uni-directional link failures and to improve the speed at which the device detects a failure in the link. Use UDLD and protected link groups simultaneously when the FastIron X Series device is connected to a device with slower link detection times.

NOTE: When UDLD and protected links are configured on a port and the link goes down, protected links will not come up after UDLD becomes “healthy” again without first physically disabling then re-enabling the link.

Configuration Notes

- You can configure a maximum of 32 protected link groups.
- There is no restriction on the number of ports in a protected link group.
- Each port can belong to one protected link group at a time.
- There is no restriction on the type of ports in a protected link group. A protected link group can consist of 10-GbE ports, Gigabit fiber ports, 10/100/1000 copper ports, and 10/100 ports, or any combination thereof.

- This feature is supported with tagged and untagged ports.
- This feature is supported with trunk ports.
- There is no restriction on the properties of ports in a protected link group. For example, member ports can be in the same VLAN or in different VLANs.

Creating a Protected Link Group and Assigning an Active Port

To create a protected link group:

1. Specify the member ports in the protected link group. Enter a command such as the following:

   ```
   FastIron(config)#protected-link-group 10 e 1 to 4
   ```

2. Optionally specify which port will be the active port for the protected link group. Enter a command such as the following:

   ```
   FastIron(config)#protected-link-group 10 active-port e 1
   ```

NOTE: If you do not explicitly configure an active port, the Foundry device automatically assigns one as the first port in the protected link group to come up.

These commands configure port e1 as the active port and ports e2 – e4 as standby ports. If port 1 goes down, the Foundry device enables the first available standby port, and switches the traffic to that port. Since the above configuration consists of a statically configured active port, the active port pre-empts other ports in the protected link group. See “About Active Ports” on page 12-6.

Syntax: [no] protected-link-group <group-ID> ethernet [<slotnum>/]<portnum> to [<slotnum>/]<portnum>

The <group-ID> parameter specifies the protected link group number. Enter a number from 1 – 32.

Each **ethernet** parameter introduces a port group.

The [<slotnum>/]portnum> **to** [<slotnum>/]<portnum> parameters specify the ports in the protected link group. The <slotnum> parameter is required on chassis devices.

Syntax: [no] protected-link-group <group-ID> active-port ethernet <[slotnum/]portnum>

The <group-ID> parameter specifies the protected link group number. Enter a number from 1 – 32.

The **active-port ethernet** [<slotnum>/]portnum> parameter defines the active port. The <slotnum> parameter is required on chassis devices.

Viewing Information about Protected Link Groups

You can view protected link group information using the **show protected-link-group** command. The following shows an example output.

```
FastIron#show protected-link-group

Group ID: 1
Member Port(s): ethe 1 to 7
Configured Active Port: 7
Current Active Port: 7
Standby Port(s): ethe 5

Total Number of Protected Link Groups: 1
```

Syntax: show protected-link-group [group-ID]

Table 12.3: CLI Display of Protected Link Group Information

This Field...	Displays...
Group ID	The ID number of the protected link group.
Member Port(s)	The ports that are members of the protected link group.
Configured Active Port	The statically configured active port. If you do not statically configure an active port, this value will be "None".
Current Active Port	The current active port for the protected link group. If all member ports are down, this value will be "None".
Standby Port(s)	The member ports that are on standby.

Chapter 13

Configuring Trunk Groups and Dynamic Link Aggregation

This chapter describes how to configure trunk groups and 802.3ad link aggregation.

- ***Trunk groups*** are manually-configured aggregate links containing multiple ports.
- ***802.3ad link aggregation*** is a protocol that dynamically creates and manages trunk groups.

NOTE: You can use both types of trunking on the same device. However, you can use only one type of trunking for a given port. For example, you can configure port 1/1 as a member of a static trunk group or you can enable 802.3ad link aggregation on the port, but you cannot do both.

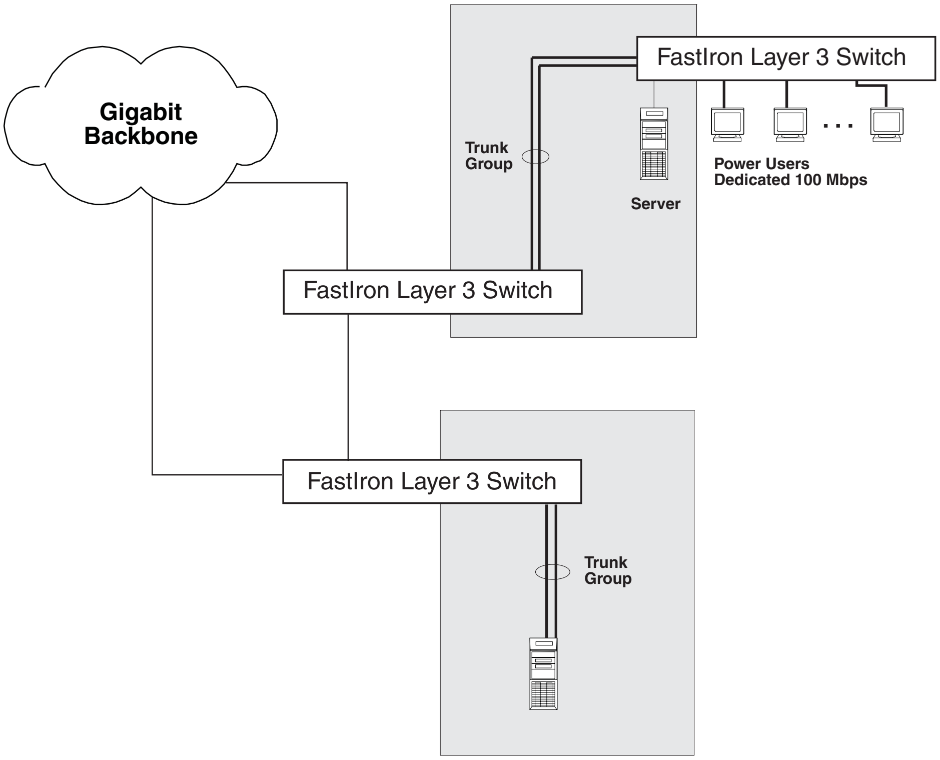
Trunk Group Overview

The Trunk Group feature allows you to manually configure multiple high-speed load-sharing links between two Foundry Layer 2 Switches or Layer 3 Switches or between a Foundry Layer 2 Switch and Layer 3 Switch and a server.

In addition to enabling load sharing of traffic, trunk groups provide redundant, alternate paths for traffic if any of the segments fail.

Figure 13.1 shows an example of a configuration that uses trunk groups.

Figure 13.1 Trunk Group Application within a FastIron Network



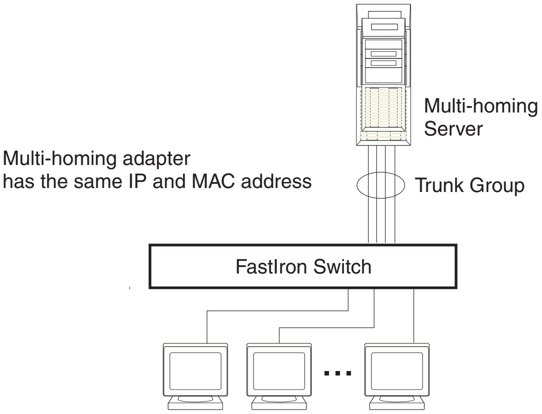
NOTE: The ports in a trunk group make a single logical link. Therefore, all the ports in a trunk group must be connected to the same device at the other end.

Trunk Group Connectivity to a Server

To support termination of a trunk group, the server must have either multiple network interface cards (NICs) or either a dual or quad interface card installed. The trunk server is designated as a server with multiple adapters or a single adapter with multiple ports that share the same MAC and IP address.

Figure 13.2 shows an example of a trunk group between a server and a Foundry device.

Figure 13.2 Trunk Group Between a Server and a Foundry Compact Layer 2 Switch or Layer 3 Switch



Trunk Group Rules

Table 13.1 lists the maximum number of trunk groups you can configure on a Foundry FastIron device and the valid number of ports in a trunk group.

Table 13.1: Trunk Group Support

Model	Maximum Number of Gigabit Trunk Groups	Valid Number of Ports in a Group
FESX424 and FWSX424	13	2, 3, or 4
FESX448 and FWSX448	25	2, 3, or 4
FGS624P and FGS624XGP	13	2, 3, 4, 5, 6, 7, or 8
FGS648P	25	2, 3, 4, 5, 6, 7, or 8
FSX	31	2, 3, or 4
FLS624	13	8
FLS648	25	8

- You cannot configure a port as a member of a trunk group if 802.3ad link aggregation is enabled on the port.
- Unlike the FES and other Foundry devices, trunk groups on the FastIron X Series, FastIron GS, and FastIron LS are not classified as switch trunk groups or server trunk groups.
- Multi-slot trunk groups are supported on GbE ports on FSX and SX chassis devices, and on 10-GbE ports starting in software release FSX 03.2.00. See “Example 2: Configuring a Trunk Group That Spans Gigabit Ethernet Modules in a Chassis Device” on page 13-11.

- Support for multi-slot trunk groups with one port per module was added in software release FSX 04.0.00. See “Example 3: Configuring a Multi-Slot Trunk Group with One Port per Module” on page 13-11.
- Starting in FGS software release 03.0.00, trunking is supported on the 1-port 10-GbE module (port 0/3/1).
- Although the FastIron devices have port ranges, they do not apply to trunk groups.
- You can select any port to be the primary port of the trunk group.
- You cannot combine Gigabit and 10-Gigabit ports in the same trunk group.
- Software release FSX 04.0.00 introduced support for one port per module in a multi-slot trunk group. Previous releases support a minimum of two ports per module. The syntax is the same as for regular multi-slot trunks, except that the port range for each slot is a single port. See “Example 3: Configuring a Multi-Slot Trunk Group with One Port per Module” on page 13-11.
- On FastIron X Series devices, port assignment on a module must be contiguous . The port range on the module cannot contain gaps. For example, you can configure ports 1, 2, 3, and 4 on a module together as trunk group, but not ports 1, 3, and 4 (excluding 2).
- Starting in FGS release 03.0.00, port assignment on a module need not be contiguous. The port range can contain gaps. For example, you can configure ports 1, 3, and 4 (excluding 2). See “Flexible Trunk Group Membership” .
- Make sure the device on the other end of the trunk link can support the same number of ports in the link. For example, if you configure a three-port trunk group on the FESX and the other end is a different type of switch, make sure the other switch can support a three-port trunk group.
- All the ports must be connected to the same device at the other end.
- All trunk group member properties must match the lead port of the trunk group with respect to the following parameters:
  - port tag type (untagged or tagged port)
  - statically configured port speed and duplex
  - QoS priority

  To change port parameters, you must change them on the primary port. The software automatically applies the changes to the other ports in the trunk group.

Flexible Trunk Group Membership

Platform Support:

- FGS and FLS devices running software release 03.0.00 and later
- FGS and FLS devices running software release 04.1.00 and later (LACP support)

FGS and FLS devices provide support for flexible trunk group membership. This feature eliminates the requirement for port membership to be consecutive, and allows the trunking of ports on non-consecutive interfaces. It also allows port 0/3/1 on the FGS626XGP to be added to a trunk group. This feature is supported on static and LACP trunk ports, as well as GbE and 10-GbE ports.

Configuration Notes

- Gigabit and 10 Gigabit ports cannot be grouped together.
- This feature is not supported from SNMP or Web management.

Configuration Syntax

For example, to configure a 4-port trunk on a FastIron GS, enter the following command:

```
FastIron(config)#trunk ethe 0/1/7 ethe 0/1/9 ethe 0/1/11 ethe 0/1/21
```

This creates a 4-port trunk group with the following members:

0/1/7, 0/1/9, 0/1/11, 0/1/21

Syntax: [no] trunk ethernet <port-list>

For <port-list>, specify the port(s) in one of the following formats:

- FastIron GS and LS compact switches – <stacknum/slotnum/portnum>
- FastIron chassis devices – <slotnum/portnum>
- FESX, and FWSX compact switches – <portnum>

You can list all of the ports individually, use the keyword **to** to specify ranges of ports, or a combination of both. To apply the configuration to all ports on the device, use the keyword **all** instead of listing the ports individually.

Users can configure up to 8 ports in a trunk group (each trunk group should have a minimum of 2 ports). Any number of members between 2 and 8 is supported for 1 Gigabit port trunk groups, and a maximum of 2 members for 10 Gigabit trunk groups.

The FLS supports the following 10 Gigabit trunk groups:

- The FLS624 supports trunks 0/2/1-0/3/1, 0/3/1-0/4/1, or 0/2/1 - 0/4/1
- The FLS648 supports trunks 0/2/1 - 0/3/1

The FGS supports the following 10 Gigabit trunk groups:

- The FGS624P supports trunks 0/2/1 - 0/2/2
- The FGS624XGP supports trunks 0/2/1 - 0/2/2, 0/2/1- 0/3/1, 0/2/2 - 0/3/1
- The FGS648P supports trunks 0/2/1 - 0/2/2

Output for many of the **show** commands will show the first and last port in a trunk as FirstPort-LastPort, if the ports are consecutive, and FirstPort*LastPort if the ports are not consecutive.

With the configuration above, output from the **show mac** command resembles the following, which shows the first and last ports.

```
FastIron#show mac
Total active entries from all ports = 1
MAC-Address Port Type Index
0007.e910.c201 0/1/7*0/1/21 Dynamic 2920
```

For a trunk group with members 0/1/7 to 0/1/9, the output from the show mac command resembles the following:

```
FastIron#show mac
Total active entries from all ports = 1
MAC-Address Port Type Index
0007.e910.c201 0/1/7-0/1/9 Dynamic 2920
```

Trunk Group Configuration Examples

Figure 13.3 shows some examples of valid 2-port and 3-port trunk group links between devices. The trunk groups in this example are switch trunk groups between two Foundry devices. Ports in a valid 2-port trunk group on one device are connected to two ports in a valid 2-port trunk group on another device. The same rules apply to 3-port, 4-port, etc., trunk groups.

Figure 13.3 Examples of 2-Port and 3-Port Trunk Groups

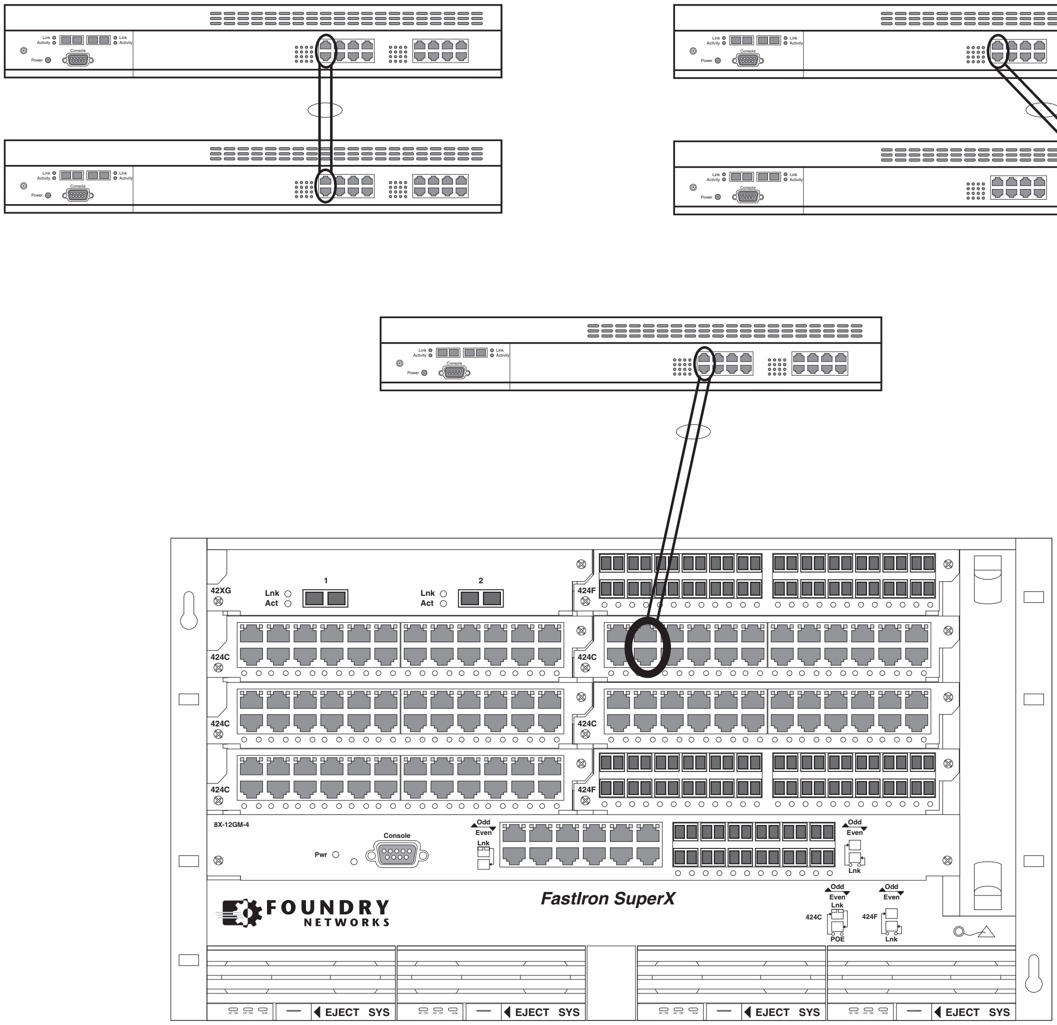
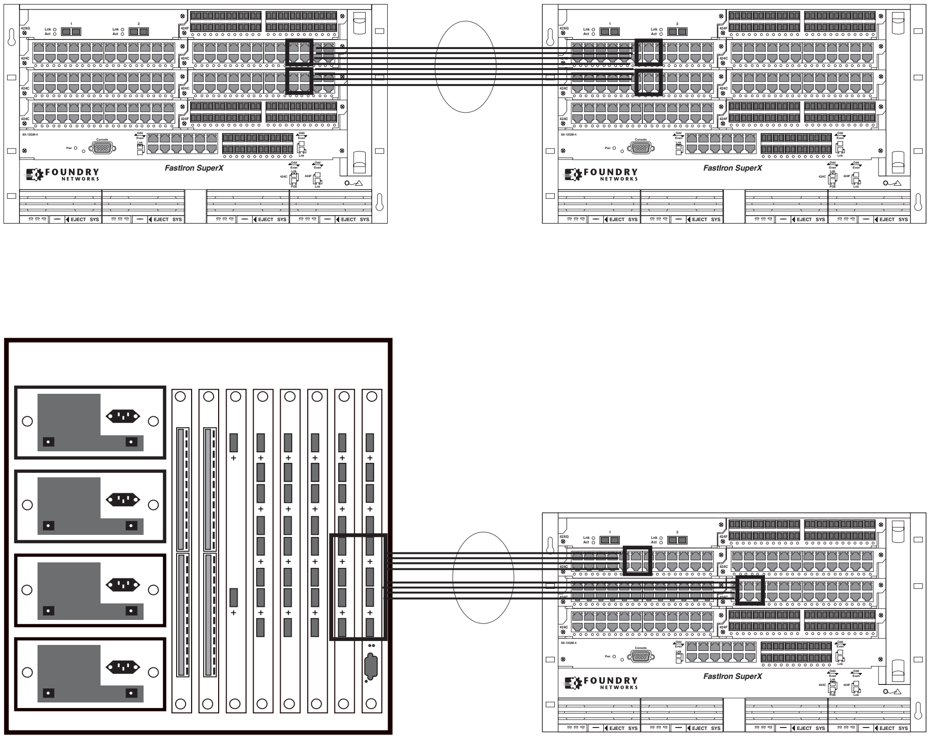


Figure 13.4 shows examples of two Chassis devices connected by multi-slot trunk groups.

Figure 13.4 Examples of Multi-Slot Trunk Groups



Trunk Group Load Sharing

Unlike other Foundry devices, trunk groups on the FastIron X Series, and FastIron GS and FastIron LS are not classified as switch trunk groups or server trunk groups.

The Foundry device load-shares across the ports in the trunk group. The method used for the load sharing depends on the following:

- Device type – Chassis device or Compact device
- Traffic type – Layer 2 or Layer 3
- Software version your device is running

NOTE: Layer 2 and Layer 3 AppleTalk traffic is not load-balanced. Layer 3 routed IP or IPX traffic also is not load balanced. These traffic types will however still be forwarded on the trunk ports.

Note Regarding IPv6

Foundry devices that support IPv6 take a packet's IPv6 address into account when sharing traffic across a trunk group. The load sharing is performed in the same way it is for IPv4 addresses; that is; trunk types whose traffic load is shared based on IPv4 address information can now use IPv6 addresses to make the load sharing decision.

Load sharing occurs as described in Table 13.3 or Table 13.2.

Load Sharing for Unknown Unicast, Multicast, and Broadcast Traffic

The Foundry device load balances unknown unicast, multicast, and broadcast traffic based on the source port and VLAN ID and not on any source or destination information in the packet.

For example, when the switch receives unknown unicast, multicast, and broadcast packets, and the packets are from the same source port, the packets are forwarded to the same port of the trunk group. Conversely, when the switch receives unknown unicast, multicast, and broadcast packets, and the packets are from different source ports, the packets are load-balanced across all the ports of the trunk group.

Note that this does not apply to known unicast traffic, which is always load balanced across all the ports of a trunk group based on the traffic's Layer 2 and Layer 3 source and destination parameters.

How Trunk Load Sharing Works

Load balancing procedures differ depending on the software version your device is running. In earlier releases, the Foundry device load balances all bridged traffic based on source and destination MAC addresses. In subsequent releases, the load balancing method for bridged traffic varies depending on the traffic type.

Table 13.2 shows how the FastIron X Series devices load balance traffic across the ports in a trunk group, if the device is running FESX/FWSX software release 02.2.00 or later or FSX software release 02.1.00 or later.

Table 13.3 shows how the FESX and FSX load balance traffic across the ports in a trunk group, if the device is running a FESX/FWSX software release prior to 02.2.00 or a FSX software release prior to 02.1.00.

Table 13.4 shows how the FastIron GS and FastIron LS load balance traffic.

NOTE: Table 13.2 and Table 13.3 do not include unknown unicast, multicast, and broadcast traffic. See “Load Sharing for Unknown Unicast, Multicast, and Broadcast Traffic”

NOTE: Load balancing on the FastIron X Series and FastIron GS devices is hardware-based.

NOTE: There is no change in load balancing for routed traffic, which is always based on the source and destination IP addresses and protocol field (not applicable for FastIron GS and FastIron LS).

Table 13.2 shows how the FastIron X Series devices load balance traffic across the ports in a trunk group, if the device is running FESX/FWSX software release 02.2.00 or later or FSX software release 02.1.00 or later.

.

Table 13.2: Trunk Group Load Sharing on FSX Devices and FESX/FWSX Devices (Later Releases)

Traffic Type	Load Balancing Method
Layer 2 bridged non-IP	Source and destination MAC addresses
Layer 2 bridged TCP/UDP	Source and destination IP addresses and Source and Destination TCP/UDP ports
Layer 2 bridged IP (non-TCP/UDP)	Source and destination IP addresses
Layer 3 routed traffic	Source and destination IP addresses and protocol field

Table 13.3 shows how the FESX and FSX load balance traffic across the ports in a trunk group, if the device is running a FESX/FWSX software release prior to 02.2.00 or a FSX software release prior to 02.1.00.

Table 13.3: Trunk Group Load Sharing on FSX Devices and FESX/FWSX Devices (Early Releases)

Traffic Layer	Traffic Type	Load-Sharing Basis
Layer 2	All traffic types	Destination MAC address Source MAC address

Table 13.3: Trunk Group Load Sharing on FSX Devices and FESX/FWSX Devices (Early Releases)

Traffic Layer	Traffic Type	Load-Sharing Basis
Layer 3	IP	Destination IP address Source MAC address Protocol
	All other traffic types	Destination MAC address Source MAC address

Table 13.4 describes how the FastIron GS load balances traffic.

Table 13.4: Trunk Group Load Sharing on FastIron GS and FastIron LS Devices

Traffic Type	Load Balancing Method
L2 Bridged Non-IP	Source MAC, Destination MAC
L2 Bridged IPv4 TCP/UDP	Source IP, Destination IP, Source TCP/UDP Port, Destination TCP/UDP Port
L2 Bridged IPv4 Non-TCP/UDP	Source IP, Destination IP
L2 Bridged IPv6 TCP/UDP	Source IP, Destination IP, Flow Label, Source TCP/UDP Port, Destination TCP/UDP Port
L2 Bridged IPv6 Non-TCP/UDP	Source IP, Destination IP, Flow Label

Adding Layer 2 Information to Trunk Hash Output

Platform Support:

- FGS and FLS devices running software release 03.0.00 and later

FGS and FLS devices support the option to include Layer 2 information in the trunk hash calculation for IP packets. Use the following new CLI command:

Syntax: [no] trunk hash-options include-layer2

This command adds Layer 2 information (text in bold) to the following load-balancing parameters:

1. Non-IP: Source MAC, Destination MAC
2. IPv4 TCP/UDP: Source IP, Destination IP, Source TCP/UDP Port, Destination TCP/UDP Port, **Source MAC, Destination MAC**
3. IPv4 Non-TCP/UDP: Source IP, Destination IP, **Source MAC, Destination MAC**
4. IPv6 TCP/UDP: Source IP, Destination IP, Flow Label, Source TCP/UDP Port, Destination TCP/UDP Port, **Source MAC, Destination MAC**
5. IPv6 Non-TCP/UDP: Source IP, Destination IP, Flow Label, **Source MAC, Destination MAC**

Configuring a Trunk Group

To configure a trunk group, do the following:

1. Disconnect the cables from those ports on both systems that will be connected by the trunk group. Do not configure the trunk groups with the cables connected.

NOTE: If you connect the cables before configuring the trunk groups and then rebooting, the traffic on the ports can create a spanning tree loop.

---

2. Configure the trunk group on one of the two Layer 2 Switches or Layer 3 Switches involved in the configuration.

NOTE: Downtime is incurred when adding a new port to a trunk group. It is suggested that you schedule the addition of ports to a trunk group to minimize downtime and its impact to the production network.

---

3. Save the configuration changes to the startup-config file.
4. Dynamically place the new trunk configuration into effect by entering the **trunk deploy** command at the global CONFIG level of the CLI.
5. If the device at the other end of the trunk group is another Layer 2 Switch or Layer 3 Switch, repeat Steps 2 – 4 for the other device.
6. When the trunk groups on both devices are operational, reconnect the cables to those ports that are now configured as trunk groups, starting with the first port (lead port) of each trunk group.
7. To verify the link is operational, use the **show trunk** command.

Example 1: Configuring the Trunk Groups Shown in Figure 13.1

To configure the trunk groups shown in Figure 13.1, enter the following commands. Notice that the commands are entered on multiple devices.

To configure the trunk group link between FSX1 and the FESX:

NOTE: The text shown in italics in the CLI example below shows messages echoed to the screen in answer to the CLI commands entered.

```
FastIron(config)#trunk e 1/5 to 1/8
Trunk will be created in next trunk deploy
FastIron(config)#write memory
FastIron(config)#trunk deploy
```

To configure the trunk group link between FSX2 and the server:

```
FastIron(config)#trunk e 1/2 to 1/4
Trunk will be created in next trunk deploy
FastIron(config)#write memory
FastIron(config)#trunk deploy
```

You then configure the trunk group on the FESX.

```
FastIron(config)#trunk ethernet 17 to 18
Trunk will be created in next trunk deploy
FastIron(config)#write memory
FastIron(config)#trunk deploy
```

NOTE: The **trunk deploy** command dynamically places trunk configuration changes into effect, without a software reload.

Example 2: Configuring a Trunk Group That Spans Gigabit Ethernet Modules in a Chassis Device

This section shows how to configure a trunk group that spans two modules in a Chassis device.

Multi-slot trunk groups are supported on 1-GbE ports, 10-GbE ports (starting In software release FSX 03.2.00), as well as on static and LACP trunk ports. Multi-slot trunk groups follow the same configuration rules listed in “Trunk Group Rules” on page 13-3.

To configure a trunk group consisting of two groups of ports, 1/1 – 1/4 on module 1 and 4/5 – 4/8 on module 4, enter the following commands:

```
FastIron(config)#trunk ethernet 1/1 to 1/4 ethernet 4/5 to 4/8
Trunk will be created in next trunk deploy
FastIron(config)#write memory
FastIron(config)#trunk deploy
```

NOTE: The **trunk deploy** command dynamically places trunk configuration changes into effect, without a software reload.

Example 3: Configuring a Multi-Slot Trunk Group with One Port per Module

Platform Support:

- FSX devices running software release 04.0.00 and later

You can select one port per module in a multi-slot trunk group. Software releases prior to 04.0.00 support a minimum of two ports per module in a multi-slot trunk group.

This feature is supported on GbE and 10-GbE ports, as well as on static and LACP trunk ports. Multi-slot trunk groups follow the same configuration rules listed in “Trunk Group Rules” on page 13-3.

To configure a two-port multi-slot trunk group consisting of ports 1/1 on module 1 and 2/1 on module 2, enter the following commands:

```
FastIron(config)#trunk ethernet 1/1 to 1/1 ethernet 2/1 to 2/1
Trunk will be created in next trunk deploy
FastIron(config)#write memory
FastIron(config)#trunk deploy
```

NOTE: The **trunk deploy** command dynamically places trunk configuration changes into effect, without a software reload.

CLI Syntax

Syntax: [no] trunk ethernet [<slotnum>/]<primary-portnum> to [<slotnum>/]<portnum> [ethernet [<slotnum>/]<primary-portnum> to [<slotnum>/]<portnum>]

Syntax: trunk deploy

Each **ethernet** parameter introduces a port group. Enter the slot number (if applicable) and the port number of the Ethernet port.

The <slotnum> parameter is required on chassis devices.

The <primary-portnum> **to** <portnum> parameters specify a port group. Notice that each port group must begin with a primary port. After you enter this command, the primary port of the first port group specified (which must be the group with the lower port numbers) becomes the primary port for the entire trunk group.

To configure a trunk group consisting of two groups of two ports each, enter commands such as the following:

```
FastIron(config)#trunk ethernet 1/1 to 1/2 ethernet 3/3 to 3/4
Trunk will be created in next trunk deploy
```

```
FastIron(config)#write memory
FastIron(config)#trunk deploy
```

Notice that the groups of ports meet the criteria for a multi-slot trunk group. Each group contains the same number of ports (two) and begins on a primary port (1/1 and 3/3).

Additional Trunking Options

The CLI contains commands for doing the following:

- Naming a trunk port
- Disabling or re-enabling a trunk port
- Deleting a trunk group
- Specifying the minimum number of ports for a trunk group

Naming a Trunk Port

To name an individual port in a trunk group, enter a command such as the following at the trunk group configuration level:

```
FastIron(config-trunk-4/1-4/4)#port-name customer1 ethernet 4/2
```

Syntax: [no] port-name <text> ethernet [<slotnum>/]<portnum>

The <text> parameter specifies the port name. The name can be up to 50 characters long.

The <slotnum> parameter is required for chassis devices.

This command assigns the name “customer1” to port 4/2 in the trunk group consisting of ports 4/1 – 4/4.

Disabling or Re-Enabling a Trunk Port

You can disable or re-enable individual ports in a trunk group. To disable an individual port in a trunk group, enter commands such as the following at the trunk group configuration level:

```
FastIron(config-trunk-4/1-4/4)#config-trunk-ind
FastIron(config-trunk-4/1-4/4)#disable ethernet 4/2
```

Syntax: [no] config-trunk-ind

Syntax: [no] disable ethernet [<slotnum>/]<portnum>

The **config-trunk-ind** command enables configuration of individual ports in the trunk group. If you do not use this command, the **disable** command will be valid only for the primary port in the trunk group and will disable all ports in the trunk group. You need to enter the **config-trunk-ind** command only once in a trunk group. After you enter the command, all applicable port configuration commands apply to individual ports only.

NOTE: If you enter **no config-trunk-ind**, all port configuration commands are removed from the individual ports and the configuration of the primary port is applied to all the ports. Also, once you enter the **no config-trunk-ind** command, the **enable**, **disable**, and **monitor** commands are valid only on the primary port and apply to the entire trunk group.

The **disable** command disables the port. The states of other ports in the trunk group are not affected.

If you have configured a name for the trunk port, you can specify the port name, as shown in the following example:

```
FastIron(config-trunk-4/1-4/4)#config-trunk-ind
FastIron(config-trunk-4/1-4/4)#disable customer1
```

Syntax: disable <portname>

To enable an individual port in a trunk group, enter commands such as the following at the trunk group configuration level:

```
FastIron(config-trunk-4/1-4/4)#config-trunk-ind
FastIron(config-trunk-4/1-4/4)#enable ethernet 4/2
```

Syntax: enable ethernet [<slotnum>/]<portnum>]

Syntax: enable <portname>

Disabling or Re-Enabling a Range or List of Trunk Ports

To disable a range of ports in a trunk group, enter commands such as the following:

```
FastIron(config)#trunk switch ethernet 2/1 to 2/4
FastIron(config-trunk-2/1-2/4)#config-trunk-ind
FastIron(config-trunk-2/1-2/4)#disable ethernet 2/3 to 2/4
```

This command disables ports 2/3 – 2/4 in trunk group 2/1 – 2/4.

To disable a list of ports, enter a command such as the following:

```
FastIron(config-trunk-2/1-2/4)#disable ethernet 2/1 ethernet 2/3 ethernet 2/4
```

This command disables ports 2/1, 2/3, and 2/4 in the trunk group.

You can specify a range and a list on the same command line. For example, to re-enable some trunk ports, enter a command such as the following:

```
FastIron(config-trunk-2/1-2/4)#enable ethernet 2/1 to 2/2 ethernet 2/4
```

Syntax: [no] disable ethernet [<slotnum>/]<portnum> [to [<slotnum>/]<portnum> | ethernet [<slotnum>/]<portnum>]

Syntax: [no] enable ethernet [<slotnum>/]<portnum> [to [<slotnum>/]<portnum> | ethernet [<slotnum>/]<portnum>]

The <slotnum> parameter is required on chassis devices.

The **to** <portnum> parameter indicates that you are specifying a range. Specify the lower port number in the range first, then **to**, then the higher port number in the range.

The <portnum> parameter specifies an individual port. You can enter this parameter multiple times to specify a list, as shown in the examples above.

Deleting a Trunk Group

To delete a trunk group, use "**no**" in front of the command you used to create the trunk group. For example, to remove one of the trunk groups configured in the examples above, enter the following command:

```
FastIron(config)#no trunk ethernet 1/1 to 1/2 ethernet 3/3 to 3/4
```

Syntax: no trunk ethernet [<slotnum>/]<portnum> to [<slotnum>/]<portnum>

The <slotnum> parameter is required on chassis devices.

Specifying the Minimum Number of Ports in a Trunk Group

Platform Support:

- FESX/FSX/FWSX devices running software release 03.0.00 and later
- FGS and FLS devices running software release 04.0.00 and later

You can configure the Foundry device to disable all of the ports in a trunk group when the number of active member ports drops below a specified threshold value. For example, if a trunk group has 4 ports, and the threshold for the trunk group is 3, then the trunk group is disabled if the number of available ports in the trunk group drops below 3. If the trunk group is disabled, then traffic is forwarded over a different link or trunk group.

For example, the following commands establish a trunk group consisting of 4 ports, then establish a threshold for this trunk group of 3 ports.

```
FastIron(config)#trunk e 3/31 to 3/34
FastIron(config-trunk-3/31-3/34)#threshold 3
```

In this example, if the number of active ports drops below 3, then all the ports in the trunk group are disabled.

Syntax: [no] threshold <number>

You can specify a threshold from 1 (the default) up to the number of ports in the trunk group.

Configuration Notes:

- The **disable module** command can be used to disable the ports on a module. However, on 10 Gigabit modules, the **disable module** command does not cause the remote connection to be dropped. If a trunk group consists of 10 Gigabit ports, and you use the **disable module** command to disable ports in the trunk group, which then causes the number of active ports in the trunk group to drop below the threshold value, the trunk group is not disabled.
- If you establish a threshold for a trunk used in conjunction with Metro Ring Protocol (MRP) on 10-Gigabit interfaces, then you must also enable Link Fault Signaling (LFS).
- If you specify a threshold for a trunk group, the other end of the trunk group must also have the same threshold configuration.

Displaying Trunk Group Configuration Information

To display configuration information for the trunk groups, use the **show trunk** command. This command displays information for configured trunk groups and operational trunk groups. A configured trunk group is one that has been configured in the software but has not been placed into operation by a reset or reboot. An operational trunk group is one that has been placed into operation by a reset or reboot.

Enter the following command at any CLI level:

```
FastIron#show trunk

Configured trunks:

Trunk ID: 1
HW Trunk ID: 1
Ports_Configured: 8
Primary Port Monitored: Jointly
Ports        1/1      1/2      1/3      1/4      1/5      1/6      1/7      1/8
Port Names   none     none     none     none     none     longna   test     none
Port_Status  enable   enable   enable   enable   disable  disable  enable   enable
Monitor      on       on       off      on       off      off      off      off
Mirror Port  3/3      3/4      N/A      3/5      N/A      N/A      N/A      N/A
Monitor Dir  both     in       N/A      out      N/A      N/A      N/A      N/A
Operational trunks:
Trunk ID: 1
HW Trunk ID: 1
Duplex: Full
Speed: 1G
Tag: No
Priority: level0
Active Ports: 6
Ports           1/1      1/2       1/3      1/4      1/5    1/6     1/7      1/8
Link_Status     active   active    active   active  down   down    active   active
LACP_Status    ready    ready     ready     expired  down   down   ready   ready
Load Sharing
 Mac Address    3        2         2        2        0      0       6        1
 IP             0        0         0        0        0      0       0        0
 Multicast      4        2         5        2        0      0       2        3
```

Syntax: show trunk [ethernet [<slotnum>/]<portnum> to [<slotnum>/]<portnum>]

The [<slotnum/> applies to chassis devices only.

NOTE: The **show trunk** command does not display any form of trunk when links are up.

Table 13.5 describes the information displayed by the **show trunk** command.

Table 13.5: CLI Trunk Group Information

This Field...	Displays...
Trunk ID	The trunk group number. The software numbers the groups in the display to make the display easy to use.
HW Trunk ID	The trunk ID.
Duplex	The mode of the port, which can be one of the following: • None – The link on the primary trunk port is down. • Full – The primary port is running in full-duplex. • Half – The primary port is running in half-duplex. **Note**: This field and the following fields apply only to operational trunk groups.
Speed	The speed set for the port. The value can be one of the following: • None – The link on the primary trunk port is down. • 10 – The port speed is 10 Mbps. • 100 – The port speed is 100 Mbps. • IG – The port speed is 1000 Mbps.
Tag	Indicates whether the ports have 802.1Q VLAN tagging. The value can be Yes or No.
Priority	Indicates the Quality of Service (QoS) priority of the ports. The priority can be a value from 0 – 7.
Active Ports	The number of ports in the trunk group that are currently active.
Ports	The ports in the trunk group.
Link_Status	The link status or each port in the trunk group.
LACP_Status	For more information about this feature, see the section “Displaying and Determining the Status of Aggregate Links” on page 13-27. • Ready - The port is functioning normally in the trunk group and is able to transmit and receive LACP packets. • Expired - The time has expired (as determined by timeout values) and the port has shut down because the port on the other side of the link has stopped transmitting packets. • Down - The port’s physical link is down.
Load Sharing	The number of traffic flows currently being load balanced on the trunk ports. All traffic exchanged within the flow is forwarded on the same trunk port. For information about trunk load sharing, see “Trunk Group Load Sharing” on page 13-7.

Dynamic Link Aggregation

Foundry software supports the IEEE 802.3ad standard for link aggregation. This standard describes the Link Aggregation Control Protocol (LACP), a mechanism for allowing ports on both sides of a redundant link to form a trunk link (aggregate link), without the need for manual configuration of the ports into trunk groups.

When you enable link aggregation on a group of Foundry ports, the Foundry ports can negotiate with the ports at the remote ends of the links to establish trunk groups.

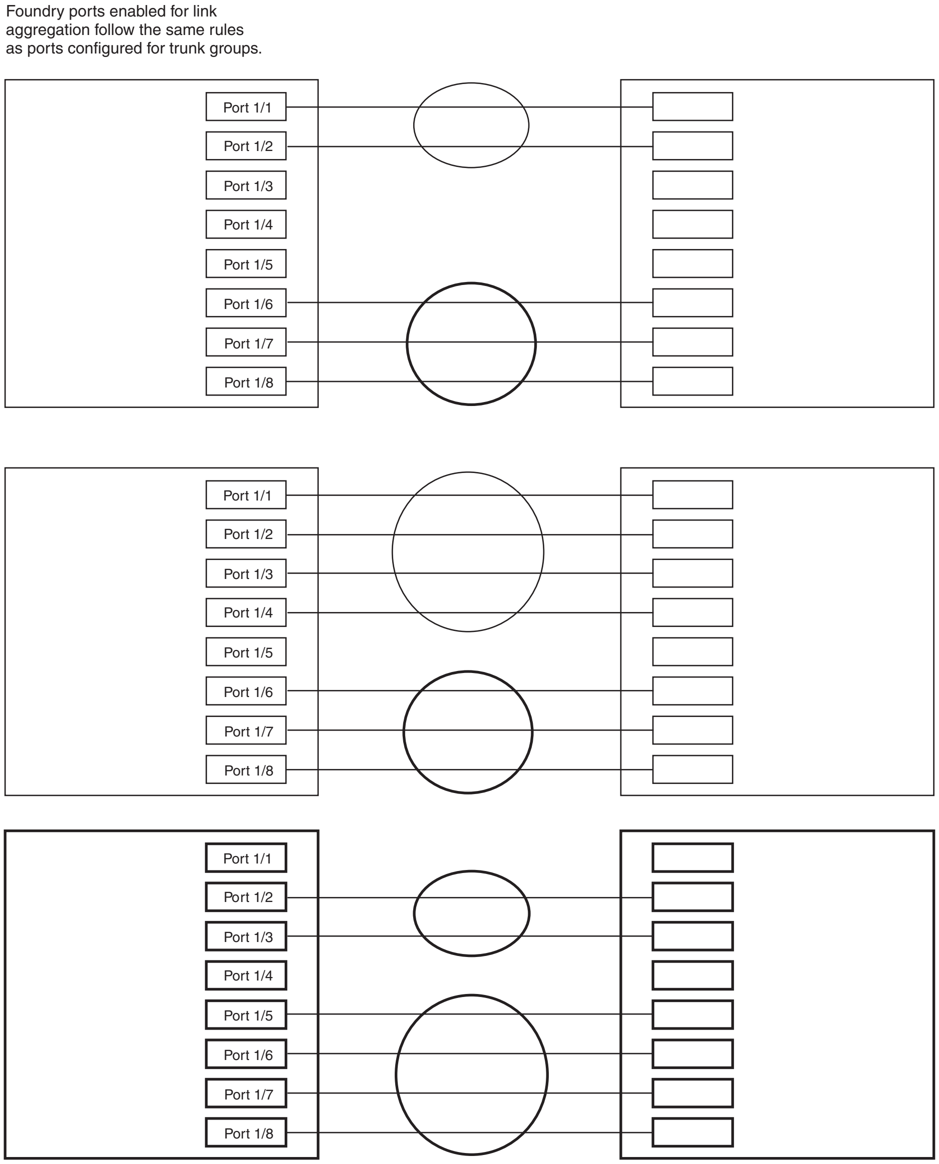
In FSX releases prior to 02.4.00, link aggregation works with untagged ports (ports that belong to a single port-based VLAN). Starting in release 02.4.00, link aggregation also works with 802.1q tagged ports (ports that belong to more than one port-based VLAN).

Configuration Example

Foundry ports follow the same configuration rules for dynamically created aggregate links as they do for statically configured trunk groups. See "Trunk Group Rules" on page 13-3 and "Trunk Group Load Sharing" on page 13-7.

Figure 13.5 on page 13-17 shows some examples of valid aggregate links.

Figure 13.5 Examples of Valid Aggregate Links


Foundry ports enabled for link aggregation follow the same rules as ports configured for trunk groups.
Port 1/1
Port 1/2
Port 1/3
Port 1/4
Port 1/5
Port 1/6
Port 1/7
Port 1/8

In this example, assume that link aggregation is enabled on all of the links between the Foundry device on the left and the device on the right (which can be either a Foundry device or another vendor's device). The ports that are members of aggregate links in this example are following the configuration rules for trunk links on Foundry devices.

The Foundry rules apply to a Foundry device even if the device at the other end is from another vendor and uses different rules. See "Trunk Group Rules" on page 13-3.

The link aggregation feature automates trunk configuration but can coexist with Foundry's trunk group feature. Link aggregation parameters do not interfere with trunk group parameters.

NOTE: Use the link aggregation feature only if the device at the other end of the link you want to aggregate also supports IEEE 802.3ad link aggregation. Otherwise, you need to manually configure the trunk links.

Link aggregation support is disabled by default. You can enable the feature on an individual port basis, in active or passive mode.

- Active mode – When you enable a port for active link aggregation, the Foundry port can exchange standard LACP Protocol Data Unit (LACPDU) messages to negotiate trunk group configuration with the port on the other side of the link. In addition, the Foundry port actively sends LACPDU messages on the link to search for a link aggregation partner at the other end of the link, and can initiate an LACPDU exchange to negotiate link aggregation parameters with an appropriately configured remote port.
- Passive mode – When you enable a port for passive link aggregation, the Foundry port can exchange LACPDU messages with the port at the remote end of the link, but the Foundry port cannot search for a link aggregation port or initiate negotiation of an aggregate link. Thus, the port at the remote end of the link must initiate the LACPDU exchange.

NOTE: Foundry recommends that you disable or remove the cables from the ports you plan to enable for dynamic link aggregation. Doing so prevents the possibility that LACP will use a partial configuration to talk to the other side of a link. A partial configuration does not cause errors, but does sometimes require LACP to be disabled and re-enabled on both sides of the link to ensure that a full configuration is used. It's easier to disable a port or remove its cable first. This applies both for active link aggregation and passive link aggregation.

Configuration Notes and Limitations

This section lists the configuration considerations and limitations for dynamic link aggregation.

FastIron GS and FastIron LS

The following notes and feature limitations apply to the FastIron GS and FastIron LS:

- The dynamic link aggregation (802.3ad) implementation on the FastIron GS and FastIron LS allows any number of ports up to eight to be aggregated into a link.
- The default key assigned to an aggregate link is based on the port type (1-Gigabit port or 10-Gigabit port). The FastIron GS assigns different keys to 10-Gigabit ports than on 1-Gigabit ports so that ports with different physical capabilities will not be able to form a trunk.

  NOTE: The trunks that will be formed by link aggregation will strictly adhere to the static trunking rules on the FastIron GS and FastIron LS. Be careful in selecting keys if you are manually configuring link aggregation keys. Make sure that the possible trunks that you expect to be formed conform to the static trunking rules.

- When you enable link aggregation (LACP) on a group of Foundry ports, you must also assign a unique key (other than the default key) to all of the ports in the aggregate link.
- 10 Gigabit links only support two port trunks.

FastIron X Series Devices

The following notes and feature limitations apply to the FastIron X Series devices:

- You cannot use 802.3ad link aggregation on a port configured as a member of a static trunk group.
- The dynamic link aggregation (802.3ad) implementation on FastIron X Series devices allows any number of ports up to four to be aggregated into a link. The feature does not require the aggregate link to consist of exactly two or four ports.
- The default key assigned to an aggregate link is based on the port type (1-Gigabit port or 10-Gigabit port). The Foundry device assigns different keys to 10-Gigabit ports than on 1-Gigabit ports so that ports with different physical capabilities will not be able to form a trunk.

  NOTE: The trunks that will be formed by link aggregation will strictly adhere to the static trunking rules on the Foundry device. Be careful in selecting keys if you are manually configuring link aggregation keys. Make sure that the possible trunks that you expect to be formed conform to the static trunking rules.

- When the feature dynamically adds or changes a trunk group, the **show trunk** command displays the trunk as both configured and active. However, the **show running-config** or **write terminal** command does not contain a trunk command defining the new or changed trunk group.
- If the feature places a port into a trunk group as a secondary port, all configuration information except information related to link aggregation is removed from the port. For example, if port 1/3 has an IP interface, and the link aggregation feature places port 1/3 into a trunk group consisting of ports 1/1 – 1/4, the IP interface is removed from the port.
- If you use this feature on a Layer 3 Switch that is running OSPF or BGP4, the feature causes these protocols to reset when a dynamic link change occurs. The reset includes ending and restarting neighbor sessions with OSPF and BGP4 peers, and clearing and relearning dynamic route entries and forwarding cache entries. Although the reset causes a brief interruption, the protocols automatically resume normal operation.
- Starting with FastIron X Series software release 02.4.00, you can enable link aggregation on 802.1Q tagged ports (ports that belong to more than one port-based VLAN). In releases prior to 02.4.00, link aggregation works with untagged ports only.

Adaptation to Trunk Disappearance

The Foundry device will tear down an aggregate link if the device at the other end of the link reboots or brings all the links down. Tearing the aggregate link down prevents a mismatch if the other device has a different trunk configuration following the reboot or re-establishment of the links. Once the other device recovers, 802.3 can renegotiate the link without a mismatch.

Flexible Trunk Eligibility

The criteria for being eligible to be in an aggregate link are flexible. A range of ports can contain down ports and still be eligible to become an aggregate link.

The device places the ports into 2-port groups by default, consisting of an odd-numbered port and the next even-numbered port. For example, ports 1/1 and 1/2 are a two-port group, as are ports 1/3 and 1/4, 9/1 and 9/10, and so on. If either of the ports in a two-port group is up, the device considers both ports to be eligible to be in an aggregate link.

Figure 13.6 shows an example of 2-port groups in a range of four ports on which link aggregation is enabled. Based on the states of the ports, some or all of them will be eligible to be used in an aggregate link.

Figure 13.6 Two-Port Groups used to Determine Aggregation Eligibility

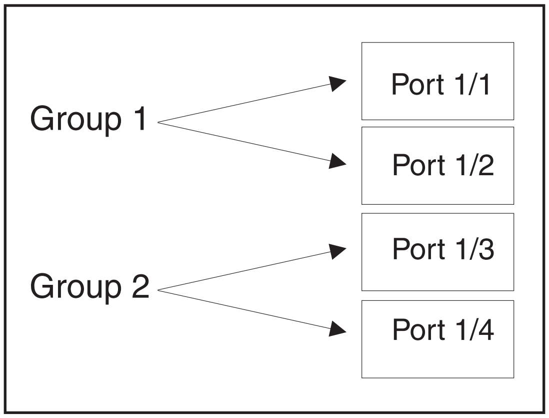


Table 13.6 shows examples of the ports from Figure 13.6 that will be eligible for an aggregate link based on individual port states.

Table 13.6: Port Eligibility for Link Aggregation

	Port Group 1		Port Group 2		Trunk Eligibility
	1/1	**1/2**	**1/3**	**1/4**	
Link State	Up	Up	Up	Up	4-port 1/1 – 1/4
	Up	Up	Up	Down	4-port 1/1 – 1/4
	Up	Down	Up	Down	4-port 1/1 – 1/4
	Up	Up	Down	Up	4-port 1/1 – 1/4
	Down	Down	Down	Up	2-port 1/3 – 1/4
	Up	Down	Down	Down	2-port 1/1 – 1/2

As shown in these examples, all or a subset of the ports within a port range will be eligible for formation into an aggregate link based on port states. Notice that the sets of ports that are eligible for the aggregate link must be valid static trunk configurations.

Command Syntax

By default, link aggregation is disabled on all ports. To enable link aggregation on a set of ports, enter commands such as the following at the interface configuration level of the CLI.

NOTE: Configuration commands for link aggregation differ depending on whether you are using the default link aggregation key automatically assigned by the software, or if you are assigning a different, unique key. Follow the commands below, according to the type of key you are using. For more information about keys, see "Key" on page 13-22.

Using the Default Key Assigned by the Software

```
FastIron(config)#interface ethernet 1/1
FastIron(config-if-e1000-1/1)#link-aggregate active
FastIron(config)#interface ethernet 1/2
FastIron(config-if-e1000-1/2)#link-aggregate active
```

The commands in this example enable the active mode of link aggregation on ports 1/1 and 1/2. The ports can send and receive LACPDU messages. Note that these ports will use the default key, since one has not been explicitly configured.

NOTE: In conformance with the 802.3ad specification, the default key assigned to an aggregate link is based on the port type (1-Gigabit port or 10-Gigabit port). The Foundry device assigns different keys to 10-Gigabit ports than 1-Gigabit ports, so that ports with different physical capabilities will not be able to form a trunk.

Assigning a Unique Key

```
FastIron(config)#interface ethernet 1/1
FastIron(config-if-e1000-1/1)#link-aggregate configure key 10000
FastIron(config-if-e1000-1/1)#link-aggregate active
FastIron(config)#interface ethernet 1/2
FastIron(config-if-e1000-1/2)#link-aggregate configure key 10000
FastIron(config-if-e1000-1/2)#link-aggregate active
```

The commands in this example assign the key 10000 and enable the active mode of link aggregation on ports 1/1 and 1/2. The ports can send and receive LACPDU messages.

NOTE: As shown in this example, when configuring a key, it is pertinent that you assign the key prior to enabling link aggregation.

The following commands enable passive link aggregation on ports 1/5 – 1/8:

```
FastIron(config)#interface ethernet 1/5 to 1/8
FastIron(config-mif-1/5-1/8)#link-aggregate passive
```

The commands in this example enable the passive mode of link aggregation on ports 1/5 – 1/8. These ports wait for the other end of the link to contact them. After this occurs, the ports can send and receive LACPDU messages.

To disable link aggregation on a port, enter a command such as the following:

```
FastIron(config-if-e1000-1/8)#link-aggregate off
```

Syntax: [no] link-aggregate active | passive | off

Syntax: [no] link-aggregate configure [system-priority <num>] | [port-priority <num>] | [key <num>]

NOTE: For more information about keys, including details about the syntax shown above, see "Key" on page 13-22.

Link Aggregation Parameters

You can change the settings on individual ports for the following link aggregation parameters:

- System priority
- Port priority
- Link type
- Singleton
- Key

System Priority

The system priority parameter specifies the Foundry device's link aggregation priority relative to the devices at the other ends of the links on which link aggregation is enabled. A higher value indicates a lower priority. You can specify a priority from 0 – 65535. The default is 1.

NOTE: If you are connecting the Foundry device to another vendor's device and the link aggregation feature is not working, set the system priority on the Foundry device to a lower priority (a higher priority value). In some cases, this change allows the link aggregation feature to operate successfully between the two devices.

Port Priority

The port priority parameter determines the active and standby links. When a group of ports is negotiating with a group of ports on another device to establish a trunk group, the Foundry port with the highest priority becomes the default active port. The other ports (with lower priorities) become standby ports in the trunk group. You can specify a priority from 0 – 65535. A higher value indicates a lower priority. The default is 1.

NOTE: This parameter is not supported in the current software release. The primary port in the port group becomes the default active port. The primary port is the lowest-numbered port in a valid trunk-port group.

Link Type

The link type parameter specifies whether the trunk is connecting to a server (server link) or to another networking device (switch link). The default link type is switch.

Timeout

You can specify a timeout mode, which determines how fast ports are removed from a trunk. You can specify a short timeout mode.

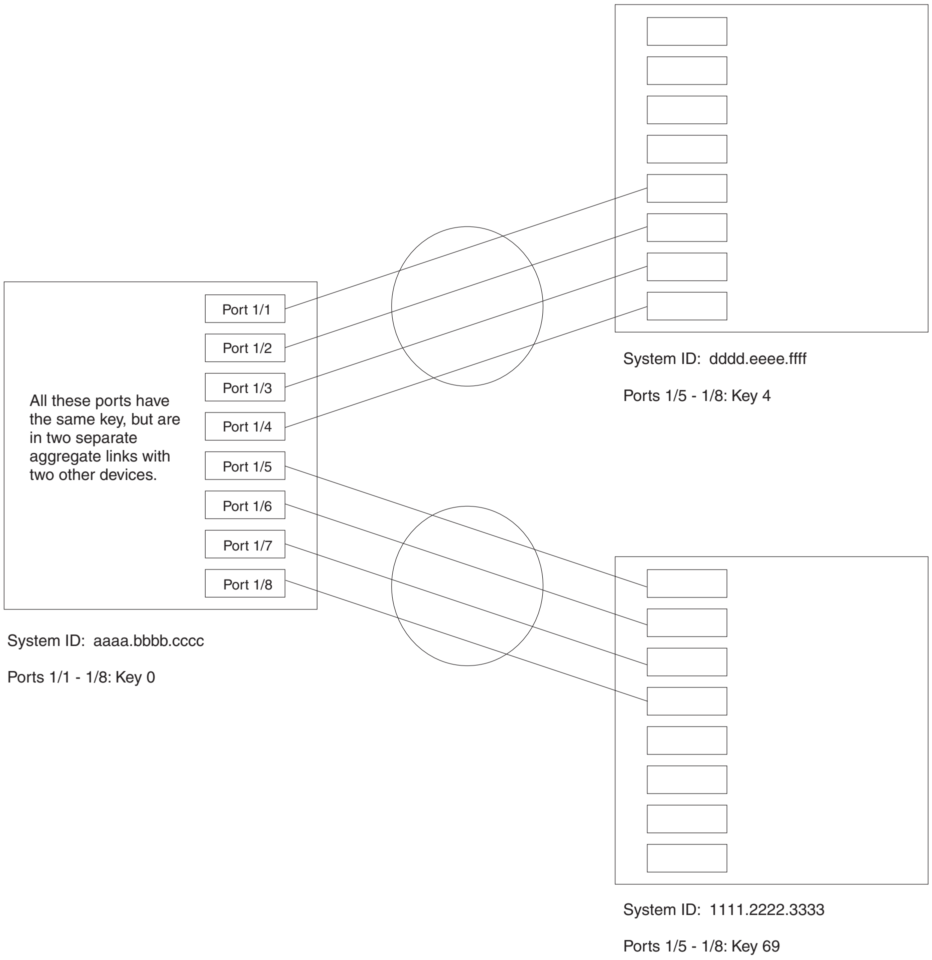
Key

Every port that is 802.3ad-enabled has a key. The key identifies the group of potential trunk ports to which the port belongs. Ports with the same key are called a key group and are eligible to be in the same trunk group.

When you enable link-aggregation on an untagged port, Foundry's software assigns a default key to the port. For tagged ports, you must manually configure link-aggregation keys. See “Configuring Keys For Ports with Link Aggregation Enabled” on page 13-26.

All ports within an aggregate link must have the same key. However, if the device has ports that are connected to two different devices, and the port groups allow the ports to form into separate aggregate links with the two devices, then each group of ports can have the same key while belonging to separate aggregate links with different devices. Figure 13.7 on page 13-23 shows an example.

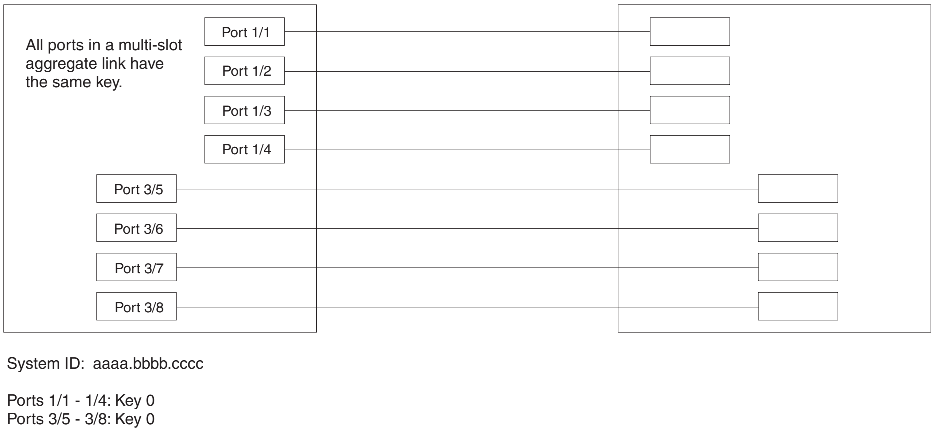
Figure 13.7 Ports with the Same Key in Different Aggregate Links


Port 1/1
Port 1/2
Port 1/3
Port 1/4
Port 1/5
Port 1/6
Port 1/7
Port 1/8
All these ports have the same key, but are in two separate aggregate links with two other devices.
System ID: aaaa.bbbb.cccc
Ports 1/1 - 1/8: Key 0
System ID: dddd.eeee.ffff
Ports 1/5 - 1/8: Key 4
System ID: 1111.2222.3333
Ports 1/5 - 1/8: Key 69

Notice that the keys between one device and another do not need to match. The only requirement for key matching is that all the ports within an aggregate link on a given device must have the same key.

Devices that support multi-slot trunk groups can form multi-slot aggregate links using link aggregation. However, the link aggregation keys for the groups of ports on each module must match. For example, if you want to allow link aggregation to form an aggregate link containing ports 1/1 – 1/4 and 3/5 – 3/8, you must change the link aggregation key on one or both groups of ports so that the key is the same on all eight ports. Figure 13.8 on page 13-24 shows an example.

Figure 13.8 Multi-Slot Aggregate Link



By default, the device's ports are divided into 4-port groups. The software dynamically assigns a unique key to each 4-port group. If you need to divide a 4-port group into two 2-port groups, change the key in one of the groups so that the two 2-port groups have different keys. For example, if you plan to use ports 1/1 and 1/2 in VLAN 1, and ports 1/3 and 1/4 in VLAN 2, change the key for ports 1/3 and 1/4.

NOTE: If you change the key for a port group, Foundry recommends that you use the value 10000 or higher, to avoid potential conflicts with dynamically created keys.

How Changing a Port's VLAN Membership Affects Trunk Groups and Dynamic Keys

When you change a port's VLAN membership and the port is currently a member of a trunk group, the following changes occur:

- The Foundry device tears down the existing trunk group.
- All ports in the trunk group get a new key.
- The new key group aggregates into a new trunk group.

When you change a port's VLAN membership, and the port is not a member of a trunk group, the following changes occur:

- The port gets a new key depending on changes to the port's VLAN tag type, as follows:
    - Tagged to Tagged VLAN – The primary port of the trunk group gets a new key.
    - Tagged to Untagged VLAN –The port gets the default key for untagged ports.
    - Untagged to Tagged VLAN – If the Foundry device finds a port with matching port properties, the port gets that port's key. If it doesn't find one, the port gets a new key.
    - Untagged to Untagged VLAN – The port gets a new key depending on whether it's in the default VLAN or not. If there is a trunk group associated with the key, it is not affected.
- All other ports keep their existing key.
- The new key groups try to aggregate into trunk groups.

Viewing Keys for Tagged Ports

To display link aggregation information, including the key for a specific port, enter a command such as the following at any level of the CLI:

```
FastIron#show link-aggregation ethernet 1/1

System ID: 00e0.52a9.bb00
Port  [Sys P] [Port P] [ Key ] [Act][Tio][Agg][Syn][Col][Dis][Def][Exp]
1/1       0        0       0   No     L    No    No    No    No    No    No
```

The command in this example shows the key and other link aggregation information for port 1/1.

To display link aggregation information, including the key for all ports on which link aggregation is enabled, enter the following command at any level of the CLI:

```
FastIron#show link-aggregation

System ID: 0004.8055.b200
Long  timeout: 90, default: 90
Short timeout: 3, default: 3

Port  [Sys P] [Port P] [ Key ] [Act][Tio][Agg][Syn][Col][Dis][Def][Exp][Ope]
1/1         1        1   10000    Yes    S    Agg   Syn  Col  Dis  Def  No   Dwn
1/2         1        1   10000    Yes    S    Agg   Syn  Col  Dis  Def  No   Dwn
2/1         1        1   10000    Yes    S    Agg   Syn  Col  Dis  Def  No   Dwn
2/2         1        1   10000    Yes    S    Agg   Syn  Col  Dis  Def  No   Dwn
4/1         1        1     480    Yes    S    Agg   Syn  Col  Dis  Def  No   Dwn
4/2         1        1     480    Yes    S    Agg   Syn  Col  Dis  Def  No   Dwn
4/3         1        1     480    Yes    S    Agg   Syn  Col  Dis  Def  No   Dwn
4/4         1        1     480    Yes    S    Agg   Syn  Col  Dis  Def  No   Dwn
4/17        1        1     481    Yes    S    Agg   Syn  Col  Dis  Def  No   Ope
4/18        1        1     481    Yes    S    Agg   Syn  Col  Dis  Def  No   Ope
4/19        1        1     481    Yes    S    Agg   Syn  Col  Dis  Def  No   Ope
4/20        1        1     481    Yes    S    Agg   Syn  Col  Dis  Def  No   Ope
```

Syntax: show link-aggregation [ethernet [<slotnum>/]<portnum>]

Possible values: N/A

Default value: N/A

Configuring Link Aggregation Parameters

You can configure one or more parameters on the same command line, and in any order.

NOTE: For key configuration only, configuration commands differ depending on whether or not link aggregation is enabled on the port(s). Follow the appropriate set of commands below, according to your system's configuration.

Configuring a Port Group Key if Link Aggregation is Disabled

Use this command sequence to change the key for ports that do not have link aggregation enabled, and for all other link aggregation parameters (i.e., system priority, port priority, and link type).

For example, to change a port group's key from the one assigned by the software to another value, enter commands such as the following:

```
FastIron(config)#interface ethernet 1/1 to 1/4
FastIron(config-mif-1/1-1/4)#link-aggregate configure key 10000
FastIron(config-mif-1/1-1/4)#interface ethernet 3/5 to 3/8
FastIron(config-mif-3/5-3/8)#link-aggregate configure key 10000
```

Configuring Keys For Ports with Link Aggregation Enabled

As shown in this command sequence, to change the key on ports that already have link aggregation enabled, you must first turn OFF link aggregation, configure the new key, then re-enable link aggregation.

```
FastIron(config)#interface ethernet 1/1 to 1/4
FastIron(config-mif-1/1-1/4)#link-aggregate off
FastIron(config-mif-1/1-1/4)#link-aggregate configure key 10000
FastIron(config-mif-1/1-1/4)#link-aggregate active
FastIron(config-mif-1/1-1/4)#interface ethernet 3/5 to 3/8
FastIron(config-mif-3/5-3/8)#link-aggregate off
FastIron(config-mif-3/5-3/8)#link-aggregate configure key 10000
FastIron(config-mif-3/5-3/8)#link-aggregate active
```

These commands change the key for ports 1/1 – 1/4 and 3/5 – 3/8 to 10000. Since all ports in an aggregate link must have the same key, the command in this example enables ports 1/1 – 1/4 and 3/5 – 3/8 to form a multi-slot aggregate link.

Syntax: [no] link-aggregate configure [system-priority <num>] | [port-priority <num>] | [key <num>]

The **system-priority** <num> parameter specifies the Foundry device's link aggregation priority. A higher value indicates a lower priority. You can specify a priority from 0 – 65535. The default is 1.

The **port-priority** <num> parameter specifies an individual port's priority within the port group. A higher value indicates a lower priority. You can specify a priority from 0 – 65535. The default is 1.

The **key** <num> parameter identifies the group of ports that are eligible to be aggregated into a trunk group. The software automatically assigns a key to each group of ports. The software assigns the keys in ascending numerical order, beginning with 0. You can change a port group's key to a value from 0 – 65535.

NOTE: If you change the key for a port group, Foundry recommends that you use the value 10000 or higher, to avoid potential conflicts with dynamically created keys.

Configuring Port Timeout

Platform Support:

- FESX/FSX/FWSX devices running software release 03.2.00 and later

You can control the time it takes to remove ports from a trunk with link aggregation enabled by configuring the link aggregated port with a "short" timeout mode. Once a port is configured with a timeout mode, it will remain in that timeout mode whether it is up or down or whether or not it is part of a trunk.

All ports in a trunk should have the same timeout mode, which is checked when link aggregation is enabled on ports.

To configure a port with a short timeout mode, enter a command such as the following:

```
FastIron(config)#interface ethernet8/1
FastIron(config-if-e1000-8/1)#link-aggregation configure timeout short
```

Syntax: [no] link-aggregate configure timeout [short]

If the timeout mode is not configured for a port and link aggregation is enabled, the port starts with a short timeout mode. Once a trunk is formed, the timeout mode is changed to the long timeout mode. The value for "long" and "short" is displayed in the output for the **show link-aggregate** command.

NOTE: Combo ports take from 5-6 seconds time to come up when enabled from a disabled state.

Displaying and Determining the Status of Aggregate Links

The **show link-aggregation** command provides the ability to view the status of dynamic links. You can determine the status of ports that are members of an aggregate link, and tell whether or not LACPCU messages are being transmitted between the ports.

The following section provides details about the events that can affect the status of ports in an aggregate link and the status of LACP messages exchanged between the ports. Later sections provide instructions for viewing these status reports.

About Blocked Ports

Foundry devices can block traffic on a port or shut down a port that is part of a trunk group or aggregate link, when a port joins a trunk group and the port on the other end of the link shuts down or stops transmitting LACP packets. Depending on the timeout value set on the port, the link aggregation information expires. If this occurs, the Foundry device shuts down the port and notifies all the upper layer protocols that the port is down.

Foundry devices can also block traffic on a port that is initially configured with link aggregation. The port is blocked until it joins a trunk group. In this case, traffic is blocked, but the port is still operational.

A port remains blocked until one of the following events occur:

- Both ports in the aggregate link have the same key
- LACP brings the port back up
- The port joins a trunk group

Displaying Link Aggregation and Port Status Information

Use the **show link-aggregation** command to determine the operational status of ports associated with aggregate links.

To display the link aggregation information for a specific port, enter a command such as the following at any level of the CLI:

```
FastIron#show link-aggregation ethernet 1/1
System ID: 00e0.52a9.bb00
Port  [Sys P] [Port P] [ Key ] [Act][Tio][Agg][Syn][Col][Dis][Def][Exp] [Ope]
1/1        0        0       0   No    L   No   No   No   No   No   No    Ope
```

The command in this example shows the link aggregation information for port 1/1.

To display the link aggregation information for all ports on which link aggregation is enabled, enter the following command at any level of the CLI:

```
FastIron#show link-aggregation

System ID: 00e0.52a9.bb00
Long  timeout: 120, default: 120 Short timeout: 3, default: 3
Port  [Sys P] [Port P] [ Key ] [Act][Tio][Agg][Syn][Col][Dis][Def][Exp][Ope]
1/1        1        1       0   No    L   Agg  Syn  No   No   Def  Exp  Ope
1/2        1        1       0   No    L   Agg  Syn  No   No   Def  Exp  Ina
1/3        1        1       0   No    L   Agg  Syn  No   No   Def  Exp  Ina
1/4        1        1       0   No    L   Agg  Syn  No   No   Def  Exp  Blo
1/5        1        1       1   No    L   Agg  No   No   No   Def  Exp  Ope
1/6        1        1       1   No    L   Agg  No   No   No   Def  Exp  Ope
1/7        1        1       1   No    L   Agg  No   No   No   Def  Exp  Dwn
1/8        1        1       1   No    L   Agg  No   No   No   Def  Exp  Dwn
```

Syntax: show link-aggregation [ethernet [<slotnum>/]<portnum>]

The <slotnum> parameter is required on chassis devices.

Use **ethernet <portnum>** to display link-aggregation information for a specific port.

NOTE: Ports that are configured as part of an aggregate link must also have the same key. For more information about assigning keys, see the section "Link Aggregation Parameters" on page 13-21.

The **show link aggregation** command shows the following information.

Table 13.7: CLI Display of Link Aggregation Information

This Field...	Displays...
System ID	Lists the base MAC address of the device. This is also the MAC address of port 1 (or 1/1).
Short timeout	Supported on systems running FSX software release 03.2.00.
Port	Lists the port number.
Sys P	Lists the system priority configured for this port.
Port P	Lists the port's link aggregation priority.
Key	Lists the link aggregation key. On devices running FSX software release 03.2.00, this column displays "singleton" if the port is configured with a Single Instance of LACP. (See "Configuring Single Link LACP" on page 13-30 for more details.
Act	Indicates the link aggregation mode, which can be one of the following: • No – The mode is passive or link aggregation is disabled (off) on the port. If link aggregation is enabled (and the mode is passive), the port can send and receive LACPDU messages to participate in negotiation of an aggregate link initiated by another port, but cannot search for a link aggregation port or initiate negotiation of an aggregate link. • Yes – The mode is active. The port can send and receive LACPDU messages.
Tio	Indicates the timeout value of the port. The timeout value can be one of the following: • L – Long. The trunk group has already been formed and the port is therefore using a longer message timeout for the LACPDU messages exchanged with the remote port. Typically, these messages are used as confirmation of the health of the aggregate link. • S – Short. The port has just started the LACPDU message exchange process with the port at the other end of the link. The S timeout value also can mean that the link aggregation information received from the remote port has expired and the ports are starting a new information exchange.
Agg	Indicates the link aggregation state of the port. The state can be one of the following: • Agg – Link aggregation is enabled on the port. • No – Link aggregation is disabled on the port.

Table 13.7: CLI Display of Link Aggregation Information (Continued)

This Field...	Displays...
Syn	Indicates the synchronization state of the port. The state can be one of the following: • No – The port is out of sync with the remote port. The port does not understand the status of the LACPDU process and is not prepared to enter a trunk link. • Syn – The port is in sync with the remote port. The port understands the status of the LACPDU message exchange process, and therefore knows the trunk group to which it belongs, the link aggregation state of the remote port, and so on.
Col	Indicates the collection state of the port, which determines whether the port is ready to send traffic over the trunk link. • Col – The port is ready to send traffic over the trunk link. • No – The port is not ready to send traffic over the trunk link.
Dis	Indicates the distribution state of the port, which determines whether the port is ready to receive traffic over the trunk link. • Dis – The port is ready to receive traffic over the trunk link. • No – The port is not ready to receive traffic over the trunk link.
Def	Indicates whether the port is using default link aggregation values. The port uses default values if it has not received link aggregation information through LACP from the port at the remote end of the link. This field can have one of the following values: • Def – The port has not received link aggregation values from the port at the other end of the link and is therefore using its default link aggregation LACP settings. • No – The port has received link aggregation information from the port at the other end of the link and is using the settings negotiated with that port.
Exp	Indicates whether the negotiated link aggregation settings have expired. The settings expire if the port does not receive an LACPDU message from the port at the other end of the link before the message timer expires. This field can have one of the following values: • Exp – The link aggregation settings this port negotiated with the port at the other end of the link have expired. The port is now using its default link aggregation settings. • No – The link aggregation values that this port negotiated with the port at the other end of the link have not expired, so the port is still using the negotiated settings.
Ope	• Ope (operational) - The port is operating normally. • Ina (inactive) - The port is inactive because the port on the other side of the link is down or has stopped transmitting LACP packets. • Blo (blocked) - The port is blocked because the adjacent port is not configured with link aggregation or because it is not able to join a trunk group. To unblock the port and bring it to an operational state, enable link aggregation on the adjacent port and ensure that the ports have the same key.

Displaying Trunk Group and LACP Status Information

Use the **show trunk** command to determine the status of LACP. See “Displaying Trunk Group Configuration Information” on page 13-14.

Clearing the Negotiated Aggregate Links Table

When a group of ports negotiates a trunk group configuration, the software stores the negotiated configuration in a table. You can clear the negotiated link aggregation configurations from the software. When you clear the information, the software does not remove link aggregation parameter settings you have configured. Only the configuration information negotiated using LACP is removed.

NOTE: The software automatically updates the link aggregation configuration based on LACPDU messages. However, clearing the link aggregation information can be useful if you are troubleshooting a configuration.

To clear the link aggregation information, enter the following command at the Privileged EXEC level of the CLI:

```
FastIron#clear link-aggregate
```

Syntax: clear link-aggregate

Configuring Single Link LACP

Platform Support:

- FESX/FSX/FWSX devices running software release 03.2.00 and later.
- FGS and FLS devices running software release 04.0.00 and later

A single instance of link aggregation (or single link LACP) can be used for unidirectional link detection. Single link LACP is based on the 802.3ad LACP protocol; but allows you to form an aggregated link with only one Ethernet port. It is the preferred method for detecting unidirectional links across multi-vendor devices, instead of link-keepalive, since it is based on a standard rather than on a proprietary solution.

Configuration Notes

- This feature is supported on 1-GbE and 10-GbE ports.
- This feature is not supported on static trunk ports.
- This feature is not supported on ports that have the **link-keepalive** command (UDLD) configured.
- Software release FSX 03.2.00 introduced support for single link LACP on 10-GbE ports. Software release FSX 04.0.00 added support for single link LACP on 1-GbE ports, as well as across modules.

CLI Syntax

To form a single link LACP, the port on both sides of the link must have LACP enabled. You can then define a single link LACP at the interface level of the device by entering the following command:

```
FastIron(config)#interface ethernet 8/1
FastIron(config-if-e1000-8/1)#link-aggregate configure singleton
```

```
Link-aggregation active
```

Syntax: [no] link-aggregate configure singleton

When single link LACP is configured, the show link aggregate command displays the following information:

```
FastIron#show link-agg
System ID: 00e0.5200.0118
Long  timeout: 120, default: 120 Short timeout: 3, default: 3
Port  [Sys P] [Port P] [  Key ] [Act][Tio][Agg][Syn][Col][Dis][Def][Exp][Ope]
2/1        1        1        1   Yes   S   Agg  Syn  No   No   Def  Exp  Ina
2/2        1        1        1   Yes   S   Agg  Syn  No   No   Def  Exp  Ina
2/3        1        1 singleton  Yes   S   Agg  Syn  No   No   Def  Exp  Ina
2/4        1        1 singleton  Yes   S   Agg  Syn  No   No   Def  Exp  Dwn
```

If **singleton** is configured on the port, the “Key” column displays “singleton”. See “ CLI Display of Link Aggregation Information” on page 13-28 to interpret the information on the displayed output.

Also, when ports are logically brought up or down while **singleton** is configured on the port, the following Syslog messages are generated:

- Logical link on interface ethernet <slot#/port#> is up.
- Logical link on interface ethernet <slot#/port#> is down.

Chapter 14
Configuring Virtual LANs (VLANs)

This chapter describes how to configure Virtual LANs (VLANs) on Foundry Layer 2 Switches and Layer 3 Switches using the CLI.

VLAN Overview

The following sections provide details about the VLAN types and features supported on the FastIron family of switches.

Types of VLANs

You can configure the following types of VLANs on Foundry devices.

- Layer 2 port-based VLAN – a set of physical ports that share a common, exclusive Layer 2 broadcast domain
- Layer 3 protocol VLANs – a subset of ports within a port-based VLAN that share a common, exclusive broadcast domain for Layer 3 broadcasts of the specified protocol type
- IP subnet VLANs – a subset of ports in a port-based VLAN that share a common, exclusive subnet broadcast domain for a specified IP subnet
- IPv6 VLANs – a subset of ports in a port-based VLAN that share a common, exclusive network broadcast domain for IPv6 packets
- IPX network VLANs – a subset of ports in a port-based VLAN that share a common, exclusive network broadcast domain for a specified IPX network
- AppleTalk cable VLANs – a subset of ports in a port-based-based VLAN that share a common, exclusive network broadcast domain for a specified AppleTalk cable range

When a Foundry device receives a packet on a port that is a member of a VLAN, the device forwards the packet based on the following VLAN hierarchy:

- If the port belongs to an IP subnet VLAN, IPX network VLAN, or AppleTalk cable VLAN and the packet belongs to the corresponding IP subnet, IPX network, or AppleTalk cable range, the device forwards the packet to all the ports within that VLAN.
- If the packet is a Layer 3 packet but cannot be forwarded as described above, but the port is a member of a Layer 3 protocol VLAN for the packet's protocol, the device forwards the packet on all the Layer 3 protocol VLAN's ports.
- If the packet cannot be forwarded based on either of the VLAN membership types listed above, but the packet can be forwarded at Layer 2, the device forwards the packet on all the ports within the receiving port's port-based VLAN.

Protocol VLANs differ from IP subnet, IPX network, and AppleTalk VLANs in an important way. Protocol VLANs accept any broadcast of the specified protocol type. An IP subnet, IPx network, or AppleTalk VLAN accepts only broadcasts for the specified IP subnet, IPX network, or AppleTalk cable range.

NOTE: Protocol VLANs are different from IP subnet, IPX network, and AppleTalk cable VLANs. A port-based VLAN cannot contain both an IP subnet, IPX network, or AppleTalk cable VLAN and a protocol VLAN for the same protocol. For example, a port-based VLAN cannot contain both an IP protocol VLAN and an IP subnet VLAN.

Layer 2 Port-Based VLANs

On all Foundry devices, you can configure port-based VLANs. A port-based VLAN is a subset of ports on a Foundry device that constitutes a Layer 2 broadcast domain.

By default, all the ports on a Foundry device are members of the default VLAN. Thus, all the ports on the device constitute a single Layer 2 broadcast domain. You can configure multiple port-based VLANs. When you configure a port-based VLAN, the device automatically removes the ports you add to the VLAN from the default VLAN.

You can configure up to 4094 port-based VLANs on a Layer 2 Switch or Layer 3 Switch. On both device types, valid VLAN IDs are 1 – 4095. You can configure up to the maximum number of VLANs within that ID range.

NOTE: VLAN IDs 4087, 4090, and 4093 are reserved for Foundry internal use only. VLAN 4094 is reserved for use by Single STP. Also, in releases prior to 04.0.00, VLAN IDs 4091 and 4092 are reserved for Foundry internal use only. Starting in release 04.0.00, if you want to use VLANs 4091 and 4092 as configurable VLANs, you can assign them to different VLAN IDs. For more information, see “Assigning Different VLAN IDs to Reserved VLANs 4091 and 4092” on page 14-15.

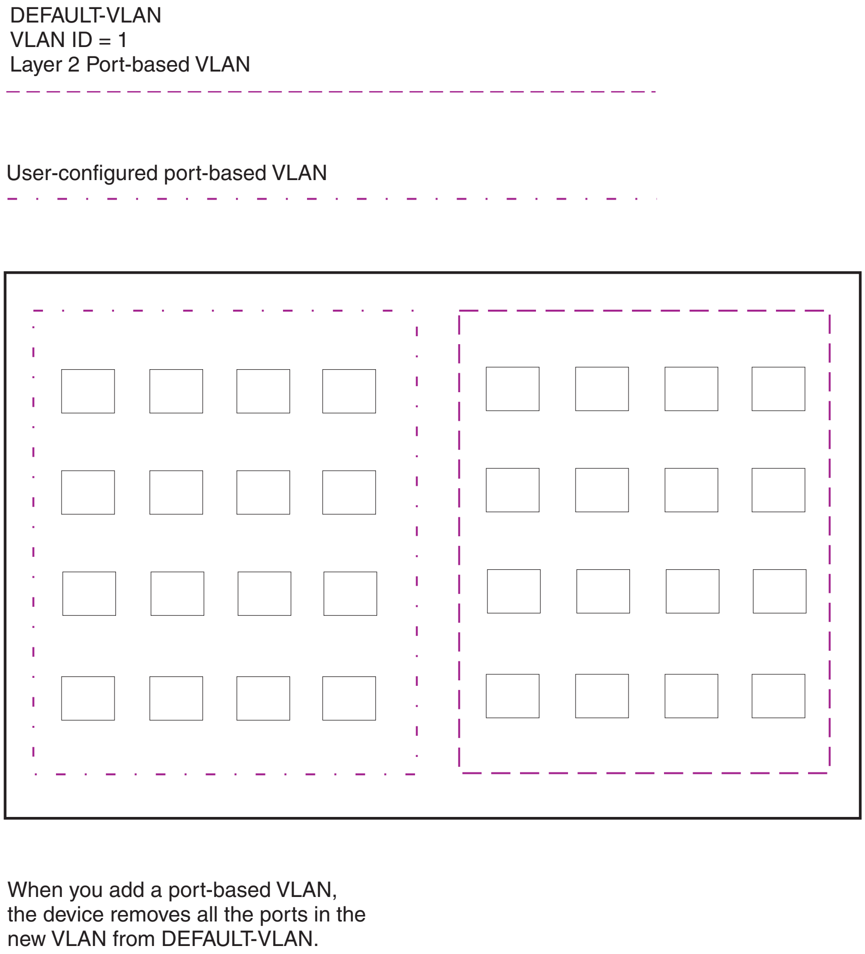
Each port-based VLAN can contain either tagged or untagged ports. A port cannot be a member of more than one port-based VLAN unless the port is tagged. ***802.1Q tagging*** allows the port to add a four-byte tag field, which contains the VLAN ID, to each packet sent on the port. You also can configure port-based VLANs that span multiple devices by tagging the ports within the VLAN. The tag enables each device that receives the packet to determine the VLAN the packet belongs to. 802.1Q tagging applies only to Layer 2 VLANs, not to Layer 3 VLANs.

Since each port-based VLAN is a separate Layer 2 broadcast domain, by default each VLAN runs a separate instance of the Spanning Tree Protocol (STP).

Layer 2 traffic is bridged within a port-based VLAN and Layer 2 broadcasts are sent to all the ports within the VLAN.

Figure 14.1 shows an example of a Foundry device on which a Layer 2 port-based VLAN has been configured.

Figure 14.1 Foundry Device Containing User-Defined Layer 2 Port-Based VLAN


DEFAULT-VLAN
VLAN ID = 1
Layer 2 Port-based VLAN
User-configured port-based VLAN
When you add a port-based VLAN, the device removes all the ports in the new VLAN from DEFAULT-VLAN.

Layer 3 Protocol-Based VLANs

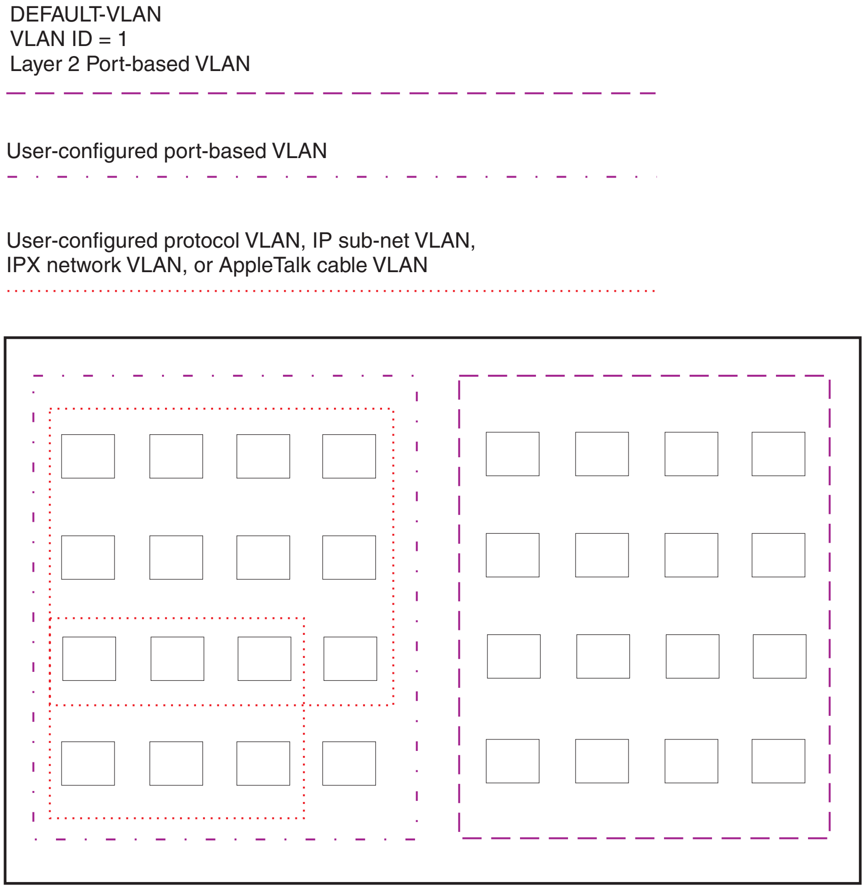
If you want some or all of the ports within a port-based VLAN to be organized according to Layer 3 protocol, you must configure a Layer 3 protocol-based VLAN within the port-based VLAN.

You can configure each of the following types of protocol-based VLAN within a port-based VLAN. All the ports in the Layer 3 VLAN must be in the same Layer 2 VLAN.

- AppleTalk – The device sends AppleTalk broadcasts to all ports within the AppleTalk protocol VLAN.
- IP – The device sends IP broadcasts to all ports within the IP protocol VLAN.
- IPv6 – The device sends IPv6 broadcasts to all ports within the IPv6 protocol VLAN.
- IPX – The device sends IPX broadcasts to all ports within the IPX protocol VLAN.
- DECnet – The device sends DECnet broadcasts to all ports within the DECnet protocol VLAN.
- NetBIOS – The device sends NetBIOS broadcasts to all ports within the NetBIOS protocol VLAN.
- Other – The device sends broadcasts for all protocol types other than those listed above to all ports within the VLAN.

Figure 14.2 shows an example of Layer 3 protocol VLANs configured within a Layer 2 port-based VLAN.

Figure 14.2 Layer 3 Protocol VLANs within a Layer 2 Port-Based VLAN


DEFAULT-VLAN
VLAN ID = 1
Layer 2 Port-based VLAN
User-configured port-based VLAN
User-configured protocol VLAN, IP sub-net VLAN,
IPX network VLAN, or AppleTalk cable VLAN

You can add Layer 3 protocol VLANs or IP sub-net, IPX network, and AppleTalk cable VLANs to port-based VLANs.

Layer 3 VLANs cannot span Layer 2 port-based VLANs.

However, Layer 3 VLANs can overlap within a Layer 2 port-based VLAN.

Integrated Switch Routing (ISR)

Foundry Networks' ***Integrated Switch Routing (ISR)*** feature enables VLANs configured on Layer 3 Switches to route Layer 3 traffic from one protocol VLAN or IP subnet, IPX network, or AppleTalk cable VLAN to another. Normally, to route traffic from one IP subnet, IPX network, or AppleTalk cable VLAN to another, you would need to forward the traffic to an external router. The VLANs provide Layer 3 broadcast domains for these protocols but do not in themselves provide routing services for these protocols. This is true even if the source and destination IP subnets, IPX networks, or AppleTalk cable ranges are on the same device.

ISR eliminates the need for an external router by allowing you to route between VLANs using virtual routing interfaces (ves). A ***virtual routing interface*** is a logical port on which you can configure Layer 3 routing parameters. You configure a separate virtual routing interface on each VLAN that you want to be able to route from or to. For example, if you configure two IP subnet VLANs on a Layer 3 Switch, you can configure a virtual routing interface on each VLAN, then configure IP routing parameters for the subnets. Thus, the Layer 3 Switch forwards IP subnet broadcasts within each VLAN at Layer 2 but routes Layer 3 traffic between the VLANs using the virtual routing interfaces.

NOTE: The Layer 3 Switch uses the lowest MAC address on the device (the MAC address of port 1 or 1/1) as the MAC address for all ports within all virtual routing interfaces you configure on the device.

The routing parameters and the syntax for configuring them are the same as when you configure a physical interface for routing. The logical interface allows the Layer 3 Switch to internally route traffic between the protocol-based VLANs without using physical interfaces.

All the ports within a protocol-based VLAN must be in the same port-based VLAN. The protocol-based VLAN cannot have ports in multiple port-based VLANs, unless the ports in the port-based VLAN to which you add the protocol-based VLAN are 802.1Q tagged.

You can configure multiple protocol-based VLANs within the same port-based VLAN. In addition, a port within a port-based VLAN can belong to multiple protocol-based VLANs of the same type or different types. For example, if you have a port-based VLAN that contains ports 1 – 10, you can configure port 5 as a member of an AppleTalk protocol VLAN, an IP protocol VLAN, and an IPX protocol VLAN, and so on.

IP Subnet, IPX Network, and AppleTalk Cable VLANs

The protocol-based VLANs described in the previous section provide separate protocol broadcast domains for specific protocols. For IP, IPX, and AppleTalk, you can provide more granular broadcast control by instead creating the following types of VLAN:

- ***IP subnet VLAN*** – An IP subnet broadcast domain for a specific IP subnet.
- ***IPX network VLAN*** – An IPX network broadcast domain for a specific IPX network.
- ***AppleTalk cable VLAN*** – An AppleTalk broadcast domain for a specific cable range.

You can configure these types of VLANs on Layer 3 Switches only. The Layer 3 Switch sends broadcasts for the IP subnet, IPX network, or AppleTalk cable range to all ports within the IP subnet, IPX network, or AppleTalk cable VLAN at Layer 2.

The Layer 3 Switch routes packets between VLANs at Layer 3. To configure an IP subnet, IPX network, or AppleTalk cable VLAN to route, you must add a virtual routing interface to the VLAN, then configure the appropriate routing parameters on the virtual routing interface.

NOTE: The Layer 3 Switch routes packets between VLANs of the same protocol. The Layer 3 Switch cannot route from one protocol to another.

NOTE: IP subnet VLANs are not the same thing as IP protocol VLANs. An IP protocol VLAN sends all IP broadcasts on the ports within the IP protocol VLAN. An IP subnet VLAN sends only the IP subnet broadcasts for the subnet of the VLAN. You cannot configure an IP protocol VLAN and an IP subnet VLAN within the same port-based VLAN.

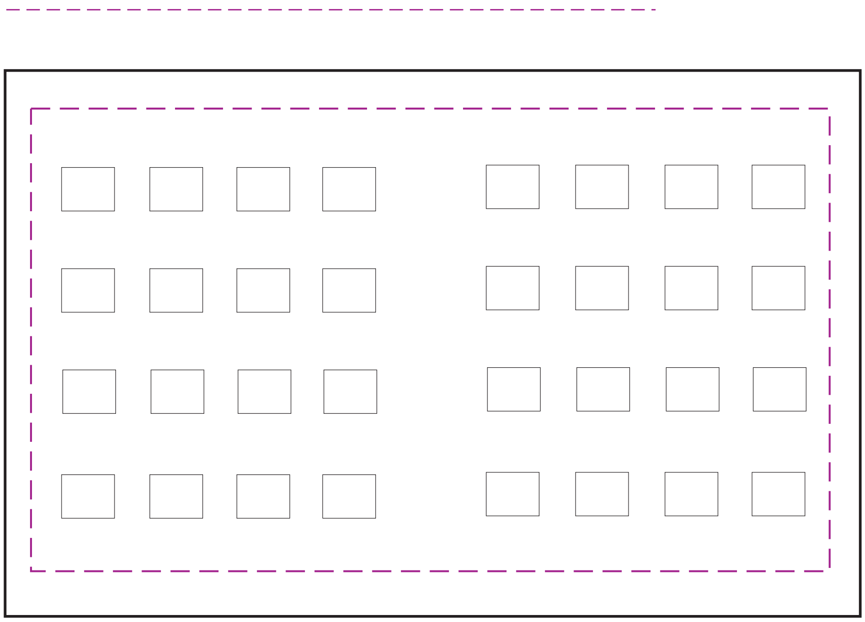
This note also applies to IPX protocol VLANs and IPX network VLANs, and to AppleTalk protocol VLANs and AppleTalk cable VLANs.

Default VLAN

By default, all the ports on a Foundry device are in a single port-based VLAN. This VLAN is called DEFAULT-VLAN and is VLAN number 1. Foundry devices do not contain any protocol VLANs or IP subnet, IPX network, or AppleTalk cable VLANs by default.

Figure 14.3 shows an example of the default Layer 2 port-based VLAN.

Figure 14.3 Default Layer 2 Port-Based VLAN



DEFAULT-VLAN
VLAN ID = 1
Layer 2 Port-based VLAN

By default, all ports belong to a single
port-based VLAN, DEFAULT-VLAN.
Thus, all ports belong to a single
Layer 2 broadcast domain.

When you configure a port-based VLAN, one of the configuration items you provide is the ports that are in the VLAN. When you configure the VLAN, the Foundry device automatically removes the ports that you place in the VLAN from DEFAULT-VLAN. By removing the ports from the default VLAN, the Foundry device ensures that each port resides in only one Layer 2 broadcast domain.

NOTE: Information for the default VLAN is available only after you define another VLAN.

Some network configurations may require that a port be able to reside in two or more Layer 2 broadcast domains (port-based VLANs). In this case, you can enable a port to reside in multiple port-based VLANs by tagging the port. See the following section.

If your network requires that you use VLAN ID 1 for a user-configured VLAN, you can reassign the default VLAN to another valid VLAN ID. See “Assigning a Different VLAN ID to the Default VLAN” on page 14-15.

802.1Q Tagging

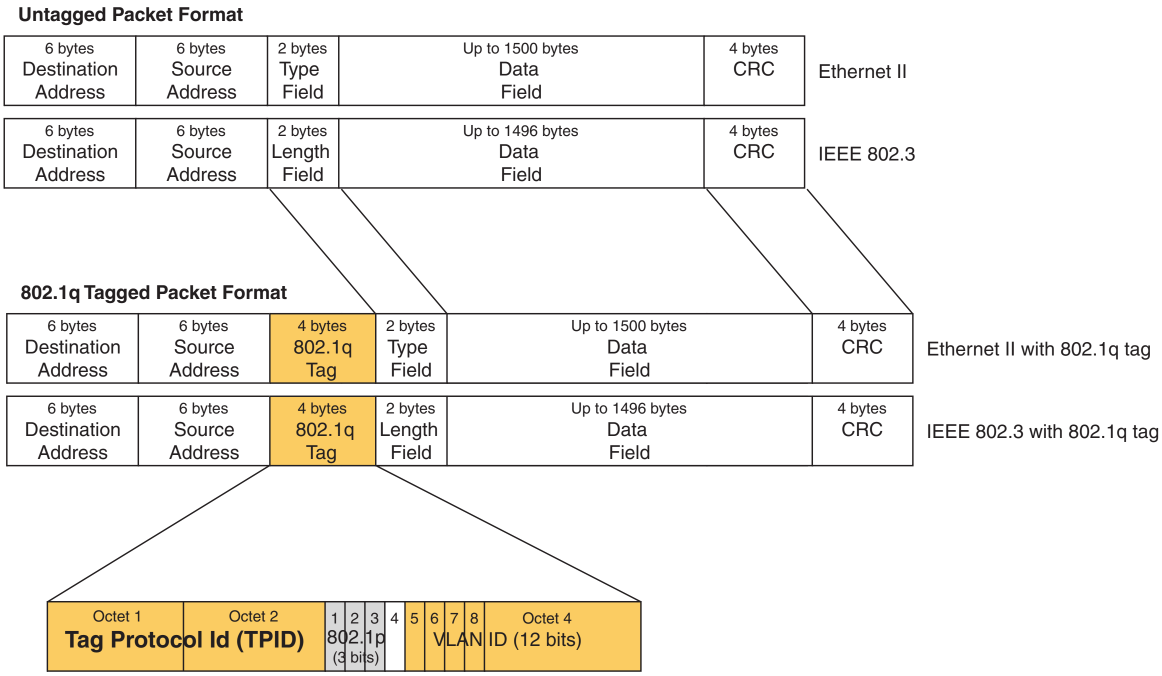
802.1Q tagging is an IEEE standard that allows a networking device to add information to a Layer 2 packet in order to identify the VLAN membership of the packet. Foundry devices tag a packet by adding a four-byte tag to the packet. The tag contains the tag value, which identifies the data as a tag, and also contains the VLAN ID of the VLAN from which the packet is sent.

- The default tag value is 8100 (hexadecimal). This value comes from the 802.1Q specification. You can change this tag value on a global basis on Foundry devices if needed to be compatible with other vendors’ equipment.

- The VLAN ID is determined by the VLAN on which the packet is being forwarded.

Figure 14.4 shows the format of packets with and without the 802.1Q tag. The tag format is vendor-specific. To use the tag for VLANs configured across multiple devices, make sure all the devices support the same tag format.

Figure 14.4 Packet Containing Foundry's 802.1QVLAN Tag

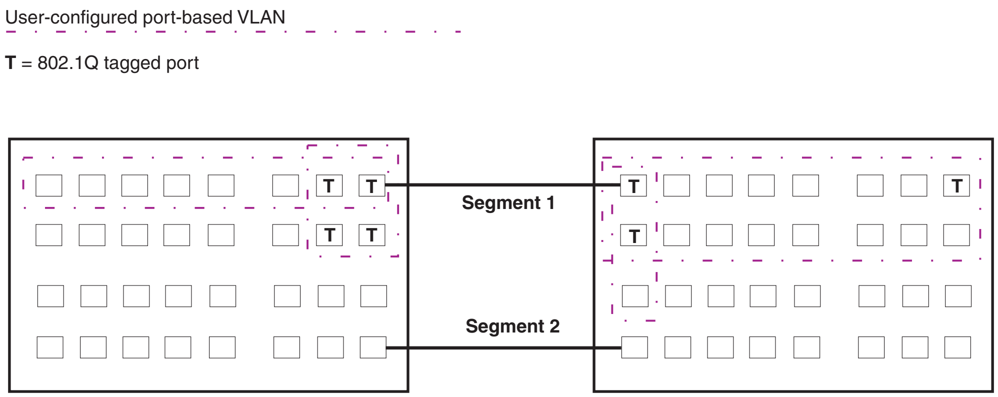


If you configure a VLAN that spans multiple devices, you need to use tagging only if a port connecting one of the devices to the other is a member of more than one port-based VLAN. If a port connecting one device to the other is a member of only a single port-based VLAN, tagging is not required.

If you use tagging on multiple devices, each device must be configured for tagging and must use the same tag value. In addition, the implementation of tagging must be compatible on the devices. The tagging on all Foundry devices is compatible with other Foundry devices.

Figure 14.5 shows an example of two devices that have the same Layer 2 port-based VLANs configured across them. Notice that only one of the VLANs requires tagging.

Figure 14.5 VLANs Configured across Multiple Devices



Segment 1

Tagging is required for the ports on Segment 1 because the ports are in multiple port-based VLANs.

Without tagging, a device receiving VLAN traffic from the other device would not be sure which VLAN the traffic is for.

Segment 2

Tagging is not required for the ports on Segment 2 because each port is in only one port-based VLAN.

802.1Q-in-Q Tagging

Foundry devices provide finer granularity for configuring 802.1Q tagging, enabling you to configure 802.1Q tag-types on a group of ports, thereby enabling the creation of two identical 802.1Q tags (802.1Q-in-Q tagging) on a single device. This enhancement improves SAV interoperability between Foundry devices and other vendors' devices that support the 802.1Q tag-types, but are not very flexible with the tag-types they accept.

For example applications and configuration details, see "Configuring 802.1Q-in-Q Tagging" on page 14-50.

Spanning Tree Protocol (STP)

The default state of STP depends on the device type:

- STP is disabled by default on Foundry Layer 3 Switches.
- STP is enabled by default on Foundry Layer 2 Switches.

Also by default, each port-based VLAN has a separate instance of STP. Thus, when STP is globally enabled, each port-based VLAN on the device runs a separate spanning tree.

You can enable or disable STP on the following levels:

- Globally – Affects all ports on the device.

NOTE: If you configure a port-based VLAN on the device, the VLAN has the same STP state as the default STP state on the device. Thus, on Layer 2 Switches, new VLANs have STP enabled by default. On Layer 3 Switches, new VLANs have STP disabled by default. You can enable or disable STP in each VLAN separately. In addition, you can enable or disable STP on individual ports.

- Port-based VLAN – Affects all ports within the specified port-based VLAN.

STP is a Layer 2 protocol. Thus, you cannot enable or disable STP for individual protocol VLANs or for IP subnet, IPX network, or AppleTalk cable VLANs. The STP state of a port-based VLAN containing these other types of VLANs determines the STP state for all the Layer 2 broadcasts within the port-based VLAN. This is true even though Layer 3 protocol broadcasts are sent on Layer 2 within the VLAN.

It is possible that STP will block one or more ports in a protocol VLAN that uses a virtual routing interface to route to other VLANs. For IP protocol and IP subnet VLANs, even though some of the physical ports of the virtual routing interface are blocked, the virtual routing interface can still route so long as at least one port in the virtual routing interface's protocol VLAN is not blocked by STP.

If you enable Single STP (SSTP) on the device, the ports in all VLANs on which STP is enabled become members of a single spanning tree. The ports in VLANs on which STP is disabled are excluded from the single spanning tree.

For more information, see "Configuring Spanning Tree Protocol (STP) Related Features" on page 9-1.

Virtual Routing Interfaces

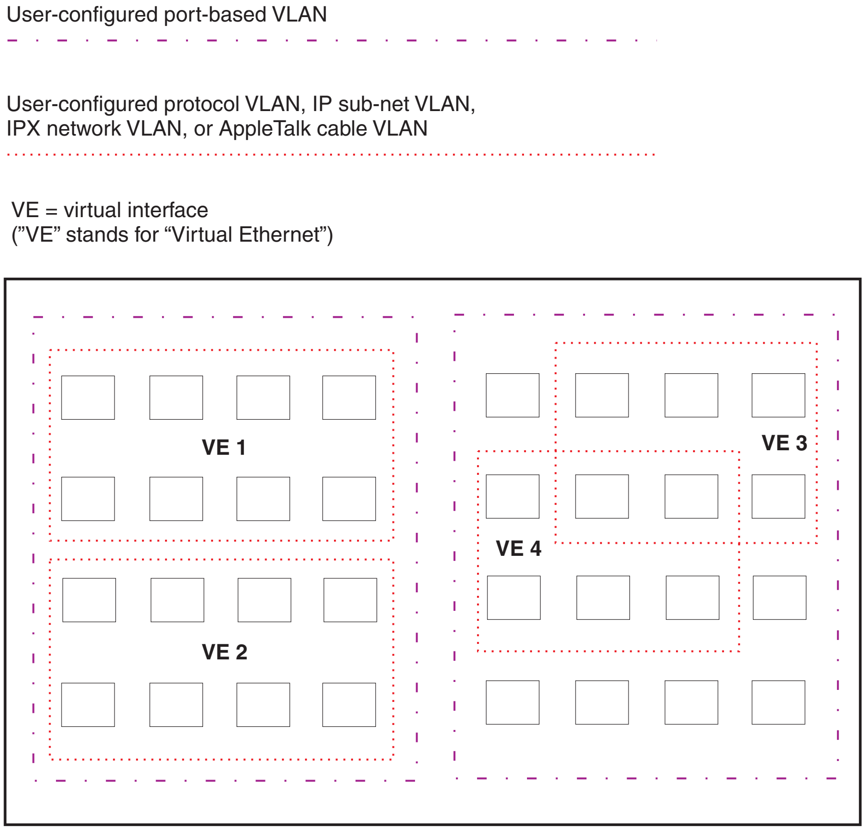
A virtual routing interface is a logical routing interface that Foundry Layer 3 Switches use to route Layer 3 protocol traffic between protocol VLANs.

Foundry devices send Layer 3 traffic at Layer 2 within a protocol VLAN. However, Layer 3 traffic from one protocol VLAN to another must be routed.

If you want the device to be able to send Layer 3 traffic from one protocol VLAN to another, you must configure a virtual routing interface on each protocol VLAN, then configure routing parameters on the virtual routing interfaces. For example, to enable a FastIron Layer 3 Switch to route IP traffic from one IP subnet VLAN to another, you must configure a virtual routing interface on each IP subnet VLAN, then configure the appropriate IP routing parameters on each of the virtual routing interfaces.

Figure 14.6 shows an example of Layer 3 protocol VLANs that use virtual routing interfaces for routing.

Figure 14.6 Use Virtual Routing Interfaces for Routing between Layer 3 Protocol VLANs


User-configured port-based VLAN
User-configured protocol VLAN, IP sub-net VLAN,
IPX network VLAN, or AppleTalk cable VLAN
VE = virtual interface
("VE" stands for "Virtual Ethernet")
VE 1
VE 2
VE 3
VE 4

Layer 2 and Layer 3 traffic within a VLAN is bridged at Layer 2.

Layer 3 traffic between protocol VLANs is routed using virtual interfaces (VE). To route to one another, each protocol VLAN must have a virtual interface.

VLAN and Virtual Routing Interface Groups

To simplify configuration, you can configure VLAN groups and virtual routing interface groups. When you create a VLAN group, the VLAN parameters you configure for the group apply to all the VLANs within the group. Additionally, you can easily associate the same IP subnet interface with all the VLANs in a group by configuring a virtual routing interface group with the same ID as the VLAN group.

For configuration information, see “Configuring VLAN Groups and Virtual Routing Interface Groups” on page 14-40.

Dynamic, Static, and Excluded Port Membership

When you add ports to a protocol VLAN, IP subnet VLAN, IPX network VLAN, or AppleTalk cable VLAN, you can add them dynamically or statically:

- Dynamic ports
- Static ports

You also can explicitly exclude ports.

Dynamic Ports

Dynamic ports are added to a VLAN when you create the VLAN. However, if a dynamically added port does not receive any traffic for the VLAN’s protocol within ten minutes, the port is removed from the VLAN. However, the port remains a candidate for port membership. Thus, if the port receives traffic for the VLAN’s protocol, the device adds the port back to the VLAN.

After the port is added back to the VLAN, the port can remain an active member of the VLAN up to 20 minutes without receiving traffic for the VLAN’s protocol. If the port ages out, it remains a candidate for VLAN membership and is added back to the VLAN when the VLAN receives protocol traffic. At this point, the port can remain in the VLAN up to 20 minutes without receiving traffic for the VLAN’s protocol, and so on.

Unless you explicitly add a port statically or exclude a port, the port is a dynamic port and thus can be an active member of the VLAN, depending on the traffic it receives.

NOTE: You cannot configure dynamic ports in an AppleTalk cable VLAN. The ports in an AppleTalk cable VLAN must be static. However, ports in an AppleTalk protocol VLAN can be dynamic or static.

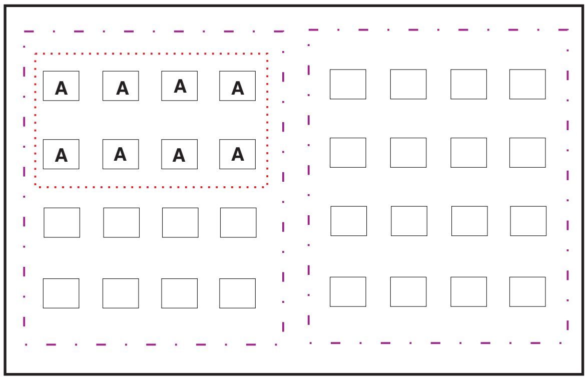
Figure 14.7 shows an example of a VLAN with dynamic ports. Dynamic ports not only join and leave the VLAN according to traffic, but also allow some broadcast packets of the specific protocol to “leak” through the VLAN. See “Broadcast Leaks” on page 14-12.

Figure 14.7 VLAN with Dynamic Ports—All Ports are Active When You Create the VLAN

A = active port

C = candidate port

When you add ports dynamically,
all the ports are added when you add
the VLAN.

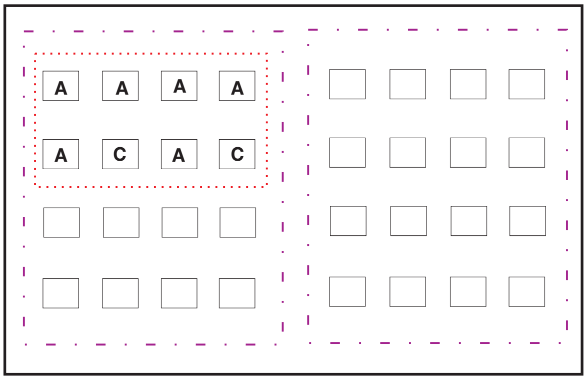


SUBNET Ports in a new protocol VLAN that do not receive traffic for the VLAN’s protocol age out after 10 minutes and become candidate ports. Figure 14.8 shows what happens if a candidate port receives traffic for the VLAN’s protocol.

Figure 14.8 VLAN with Dynamic Ports—Candidate Ports Become Active Again if they Receive Protocol Traffic

Ports that time out remain candidates for membership in the VLAN and become active again if they receive traffic for the VLAN’s protocol, IP sub-net, IPX network, or AppleTalk cable range.

When a candidate port rejoins a VLAN, the timeout for that port becomes 20 minutes. Thus, the port remains an active member of the VLAN even if it does not receive traffic for 20 minutes. After that, the port becomes a candidate port again.



Static Ports

Static ports are permanent members of the protocol VLAN. The ports remain active members of the VLAN regardless of whether the ports receive traffic for the VLAN’s protocol. You must explicitly identify the port as a static port when you add it to the VLAN. Otherwise, the port is dynamic and is subject to aging out.

Excluded Ports

If you want to prevent a port in a port-based VLAN from ever becoming a member of a protocol, IP subnet, IPX network, or AppleTalk cable VLAN configured in the port-based VLAN, you can explicitly exclude the port. You exclude the port when you configure the protocol, IP subnet, IPX network, or AppleTalk cable VLAN.

Excluded ports do not leak broadcast packets. See “Broadcast Leaks” on page 14-12.

Broadcast Leaks

A dynamic port becomes a member of a Layer 3 protocol VLAN when traffic from the VLAN's protocol is received on the port. After this point, the port remains an active member of the protocol VLAN, unless the port does not receive traffic from the VLAN's protocol for 20 minutes. If the port does not receive traffic for the VLAN's protocol for 20 minutes, the port ages out and is no longer an active member of the VLAN.

To enable a host that has been silent for awhile to send and receive packets, the dynamic ports that are currently members of the Layer 3 protocol VLAN "leak" Layer 3 broadcast packets to the ports that have aged out. When a host connected to one of the aged out ports responds to a leaked broadcast, the port is added to the protocol VLAN again.

To "leak" Layer 3 broadcast traffic, an active port sends 1/8th of the Layer 3 broadcast traffic to the inactive (aged out) ports.

Static ports do not age out and do not leak broadcast packets.

Super Aggregated VLANs

You can aggregate multiple VLANs within another VLAN. This feature allows you to construct Layer 2 paths and channels. This feature is particularly useful for Virtual Private Network (VPN) applications in which you need to provide a private, dedicated Ethernet connection for an individual client to transparently reach its subnet across multiple networks.

For an application example and configuration information, see “Configuring Super Aggregated VLANs” on page 14-43.

Trunk Group Ports and VLAN Membership

A trunk group is a set of physical ports that are configured to act as a single physical interface. Each trunk group’s port configuration is based on the configuration of the lead port, which is the lowest numbered port in the group.

If you add a trunk group’s lead port to a VLAN, all of the ports in the trunk group become members of that VLAN.

Summary of VLAN Configuration Rules

A hierarchy of VLANs exists between the Layer 2 and Layer 3 protocol-based VLANs:

- Port-based VLANs are at the lowest level of the hierarchy.
- Layer 3 protocol-based VLANs, IP, IPv6, IPX, AppleTalk, Decnet, and NetBIOS are at the middle level of the hierarchy.
- IP subnet, IPX network, and AppleTalk cable VLANs are at the top of the hierarchy.

NOTE: You cannot have a protocol-based VLAN and a subnet or network VLAN of the same protocol type in the same port-based VLAN. For example, you can have an IPX protocol VLAN and IP subnet VLAN in the same port-based VLAN, but you cannot have an IP protocol VLAN and an IP subnet VLAN in the same port-based VLAN, nor can you have an IPX protocol VLAN and an IPX network VLAN in the same port-based VLAN.

As a Foundry device receives packets, the VLAN classification starts from the highest level VLAN first. Therefore, if an interface is configured as a member of both a port-based VLAN and an IP protocol VLAN, IP packets coming into the interface are classified as members of the IP protocol VLAN because that VLAN is higher in the VLAN hierarchy.

Multiple VLAN Membership Rules

- A port can belong to multiple, unique, overlapping Layer 3 protocol-based VLANs without VLAN tagging.
- A port can belong to multiple, overlapping Layer 2 port-based VLANs only if the port is a tagged port. Packets sent out of a tagged port use an 802.1Q-tagged frame.
- When both port and protocol-based VLANs are configured on a given device, all protocol VLANs must be strictly contained within a port-based VLAN. A protocol VLAN cannot include ports from multiple port-based VLANs. This rule is required to ensure that port-based VLANs remain loop-free Layer 2 broadcast domains.
- IP protocol VLANs and IP subnet VLANs cannot operate concurrently on the system or within the same port-based VLAN.
- IPX protocol VLANs and IPX network VLANs cannot operate concurrently on the system or within the same port-based VLAN.
- If you first configure IP and IPX protocol VLANs before deciding to partition the network by IP subnet and IPX network VLANs, then you need to delete those VLANs before creating the IP subnet and IPX network VLANs.
- One of each type of protocol VLAN is configurable within each port-based VLAN on the Layer 2 Switch.
- Multiple IP subnet and IPX network VLANs are configurable within each port-based VLAN on the Layer 2 Switch.

- Removing a configured port-based VLAN from a Foundry Networks Layer 2 Switch or Layer 3 Switch automatically removes any protocol-based VLAN, IP subnet VLAN, AppleTalk cable VLAN, or IPX network VLAN, or any Virtual Ethernet router interfaces defined within the Port-based VLAN.

Routing Between VLANs

Foundry Layer 3 Switches can locally route IP, IPX, and Appletalk between VLANs defined within a single router. All other routable protocols or protocol VLANs (for example, DecNet) must be routed by another external router capable of routing the protocol.

Virtual Routing Interfaces (Layer 3 Switches Only)

You need to configure virtual routing interfaces if an IP, IPX, or Appletalk protocol VLAN, IP subnet VLAN, AppleTalk cable VLAN, or IPX network VLAN needs to route protocols to another port-based VLAN on the same router. A virtual routing interface can be associated with the ports in only a single port-based VLAN. Virtual router interfaces must be defined at the highest level of the VLAN hierarchy.

If you do not need to further partition the port-based VLAN by defining separate Layer 3 VLANs, you can define a single virtual routing interface at the port-based VLAN level and enable IP, IPX, and Appletalk routing on a single virtual routing interface.

Bridging and Routing the Same Protocol Simultaneously on the Same Device (Layer 3 Switches Only)

Some configurations may require simultaneous switching and routing of the same single protocol across different sets of ports on the same router. When IP, IPX, or Appletalk routing is enabled on a Foundry Layer 3 Switch, you can route these protocols on specific interfaces while bridging them on other interfaces. In this scenario, you can create two separate backbones for the same protocol, one bridged and one routed.

To bridge IP, IPX, or Appletalk at the same time these protocols are being routed, you need to configure an IP protocol, IP subnet, IPX protocol, IPX network, or Appletalk protocol VLAN and not assign a virtual routing interface to the VLAN. Packets for these protocols are bridged or switched at Layer 2 across ports on the router that are included in the Layer 3 VLAN. If these VLANs are built within port-based VLANs, they can be tagged across a single set of backbone fibers to create separate Layer 2 switched and Layer 3 routed backbones for the same protocol on a single physical backbone.

Routing Between VLANs Using Virtual Routing Interfaces (Layer 3 Switches Only)

Foundry calls the ability to route between VLANs with virtual routing interfaces ***Integrated Switch Routing (ISR)***. There are some important concepts to understand before designing an ISR backbone.

Virtual router interfaces can be defined on port-based, IP protocol, IP subnet, IPX protocol, IPX network, AppleTalk protocol, and AppleTalk cable VLANs.

To create any type of VLAN on a Foundry Layer 3 Switch, Layer 2 forwarding must be enabled. When Layer 2 forwarding is enabled, the Layer 3 Switch becomes a Switch on all ports for all non-routable protocols.

If the router interfaces for IP, IPX, or AppleTalk are configured on physical ports, then routing occurs independent of the Spanning Tree Protocol (STP). However, if the router interfaces are defined for any type VLAN, they are virtual routing interfaces and are subject to the rules of STP.

If your backbone consists of virtual routing interfaces all within the same STP domain, it is a bridged backbone, not a routed one. This means that the set of backbone interfaces that are blocked by STP will be blocked for routed protocols as well. The routed protocols will be able to cross these paths only when the STP state of the link is FORWARDING. This problem is easily avoided by proper network design.

When designing an ISR network, pay attention to your use of virtual routing interfaces and the spanning-tree domain. If Layer 2 switching of your routed protocols (IP, IPX, AppleTalk) is not required across the backbone, then the use of virtual routing interfaces can be limited to edge switch ports within each router. Full backbone routing can be achieved by configuring routing on each physical interface that connects to the backbone. Routing is independent of STP when configured on a physical interface.

If your ISR design requires that you switch IP, IPX, or Appletalk at Layer 2 while simultaneously routing the same protocols over a single backbone, then create multiple port-based VLANs and use VLAN tagging on the backbone links to separate your Layer 2 switched and Layer 3 routed networks.

There is a separate STP domain for each port-based VLAN. Routing occurs independently across port-based VLANs or STP domains. You can define each end of each backbone link as a separate tagged port-based VLAN. Routing will occur independently across the port-based VLANs. Because each port-based VLAN's STP domain is a single point-to-point backbone connection, you are guaranteed to never have an STP loop. STP will never block the virtual router interfaces within the tagged port-based VLAN, and you will have a fully routed backbone.

Dynamic Port Assignment (Layer 2 Switches and Layer 3 Switches)

All Switch ports are dynamically assigned to any Layer 3 VLAN on Foundry Layer 2 Switches and any non-routable VLAN on Foundry Layer 3 Switches. To maintain explicit control of the VLAN, you can explicitly exclude ports when configuring any Layer 3 VLAN on a Foundry Layer 2 Switch or any non-routable Layer 3 VLAN on a Foundry Layer 3 Switch.

If you do not want the ports to have dynamic membership, you can add them statically. This eliminates the need to explicitly exclude the ports that you do not want to participate in a particular Layer 3 VLAN.

Assigning a Different VLAN ID to the Default VLAN

When you enable port-based VLANs, all ports in the system are added to the default VLAN. By default, the default VLAN ID is “VLAN 1”. The default VLAN is not configurable. If you want to use the VLAN ID “VLAN 1” as a configurable VLAN, you can assign a different VLAN ID to the default VLAN.

To reassign the default VLAN to a different VLAN ID, enter the following command:

```
FastIron(config)#default-vlan-id 4095
```

Syntax: [no] default-vlan-d <vlan-id>

You must specify a valid VLAN ID that is not already in use. For example, if you have already defined VLAN 10, do not try to use “10” as the new VLAN ID for the default VLAN. Valid VLAN IDs are numbers from 1 – 4095.

NOTE: Changing the default VLAN name does not change the properties of the default VLAN. Changing the name allows you to use the VLAN ID “1” as a configurable VLAN.

Assigning Different VLAN IDs to Reserved VLANs 4091 and 4092

Platform Support:

- FESX/FSX/FWSX devices running software release 04.0.00 and later – L2, BL3, L3
- FGS and FLS devices running software release 04.0.00 and later

In releases prior to those listed above, VLANs 4091 and 4092 are reserved for Foundry internal VLAN management and cannot be configured. Starting in the releases listed above, if you want to use VLANs 4091 and 4092 as configurable VLANs, you can assign them to different VLAN IDs. The VLAN IDs to which the reserved VLANs are reassigned must be valid VLAN IDs that are not already in use.

For example, to reassign reserved VLAN 4091 to VLAN 10, enter the following commands:

```
FastIron(config)#reserved-vlan-map vlan 4091 new-vlan 10
Reload required.  Please write memory and then reload or power cycle.
FastIron(config)#write mem
FastIron(config)#exit
FastIron#reload
```

NOTE: You must save the configuration (write mem) and reload the software to place the change into effect.

The above configuration changes the VLAN ID of 4091 to 10. After saving the configuration and reloading the software, you can configure VLAN 4091 as you would any other VLAN.

Syntax: [no] reserved-vlan-map vlan 4091 | 4092 new-vlan <vlan-id>

For <vlan-id>, enter a valid VLAN ID that is not already in use. For example, if you have already defined VLAN 20, do not try to use “20 as the new VLAN ID. Valid VLAN IDs are numbers from 1 – 4090, 4093, and 4095. VLAN ID 4094 is reserved for use by the Single Spanning Tree feature.

Viewing Reassigned VLAN IDs for Reserved VLANs 4091 and 4092

To view the assigned VLAN IDs for reserved VLANs 4091 and 4092, use the **show reserved-vlan-map** command. The reassigned VLAN IDs also display in the output of the **show running-config** and **show config** commands.

The following shows example output for the **show reserved-vlan-map** command.

```
FastIron #show reserved-vlan-map
Reserved Purpose             Default        Re-assign        Current
  CPU VLAN                    4091             10               10
  All Ports VLAN              4092             33               33
```

Syntax: show reserved-vlan-map

The following table defines the fields in the output of the **show reserved-vlan-map** command.

Table 14.1: Output of the show reserved-vlan-map command

This field	Displays
Reserved Purpose	Describes for what the VLAN is reserved. Note that the description is for Foundry internal VLAN management.
Default	The default VLAN ID of the reserved VLAN.
Re-assign	The VLAN ID to which the reserved VLAN was reassigned.[1]
Current	The current VLAN ID for the reserved VLAN.[1]

[1] If you reassign a reserved VLAN without saving the configuration and reloading the software, the reassigned VLAN ID will display in the **Re-assign** column. However, the previously configured or default VLAN ID will display in the **Current** column until the configuration is saved and the device reloaded.

Assigning Trunk Group Ports

When a “lead” trunk group port is assigned to a VLAN, all other members of the trunk group are automatically added to that VLAN. A lead port is the first port of a trunk group port range; for example, “1” in 1 – 4 or “5” in 5 – 8. See “Trunk Group Rules” on page 13-3 for more information.

Configuring Port-Based VLANs

Port-based VLANs allow you to provide separate spanning tree protocol (STP) domains or broadcast domains on a port-by-port basis.

This section describes how to perform the following tasks for port-based VLANs using the CLI:

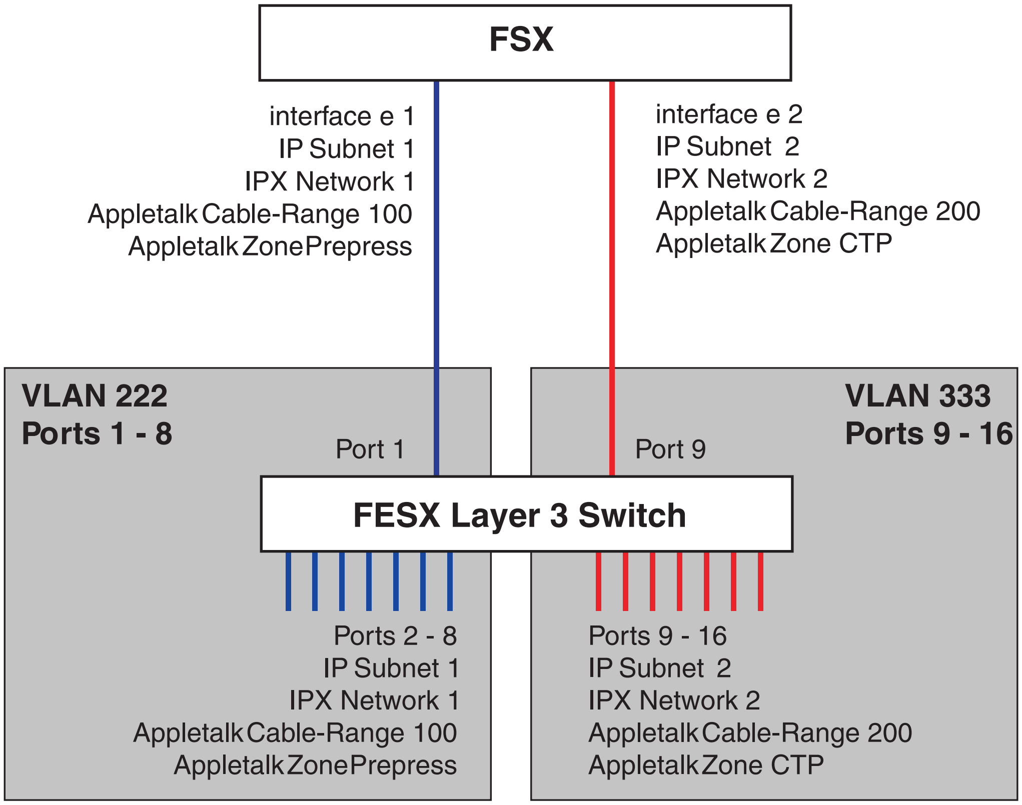
- Create a VLAN
- Delete a VLAN
- Modify a VLAN

- Change a VLAN’s priority
- Enable or disable STP on the VLAN

EXAMPLES: 1

Figure 14.9 shows a simple port-based VLAN configuration using a single Foundry Layer 2 Switch. All ports within each VLAN are untagged. One untagged port within each VLAN is used to connect the Layer 2 Switch to a Layer 3 Switch (in this example, a FSX) for Layer 3 connectivity between the two port-based VLANs.

Figure 14.9 Port-Based VLANs 222 and 333



To create the two port-based VLANs shown in Figure 14.9, enter the following commands:

```
FastIron(config)#vlan 222 by port
FastIron(config-vlan-222)#untag e 1 to 8
FastIron(config-vlan-222)#vlan 333 by port
FastIron(config-vlan-333)#untag e 9 to 16
```

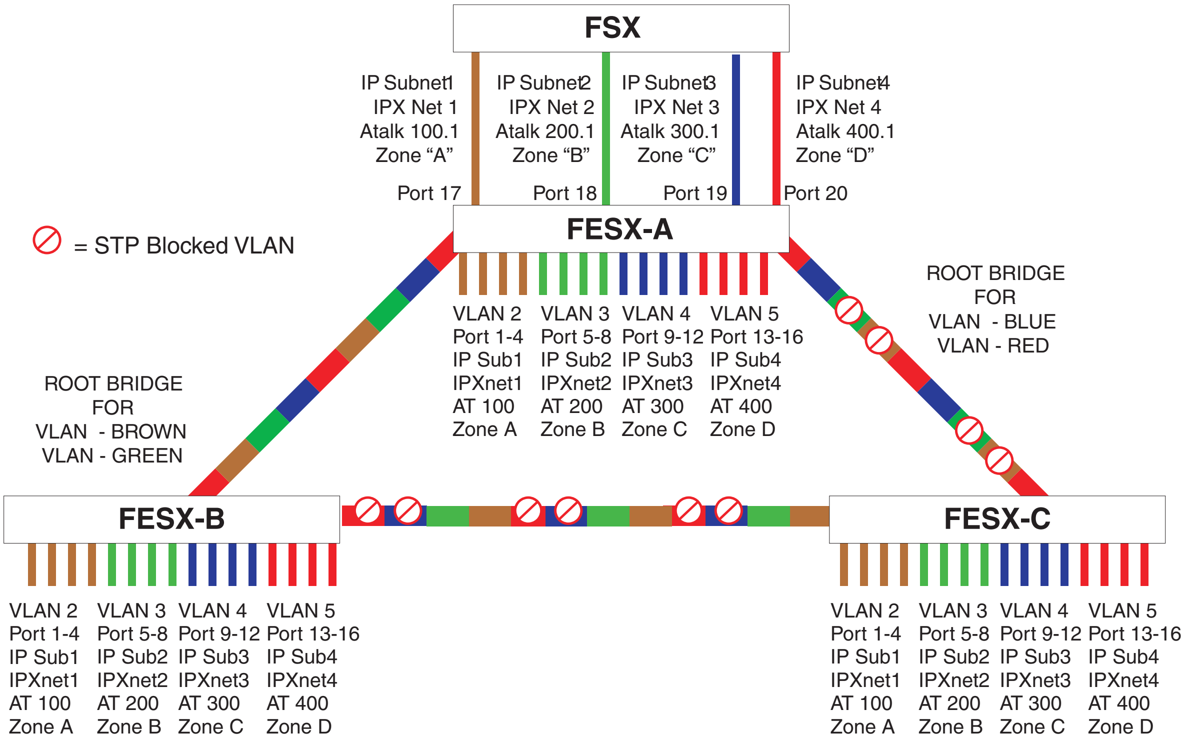
Syntax: vlan <vlan-id> by port

Syntax: untagged ethernet [<slotnum>/]<portnum> [to [<slotnum>/]<portnum> | ethernet [<slotnum>/]<portnum>]

EXAMPLES: 2

Figure 14.10 shows a more complex port-based VLAN configuration using multiple Layer 2 Switches and IEEE 802.1Q VLAN tagging. The backbone link connecting the three Layer 2 Switches is tagged. One untagged port within each port-based VLAN on FESX-A connects each separate network wide Layer 2 broadcast domain to the router for Layer 3 forwarding between broadcast domains. The STP priority is configured to force FESX-A to be the root bridge for VLANs RED and BLUE. The STP priority on FESX-B is configured so that FESX-B is the root bridge for VLANs GREEN and BROWN.

Figure 14.10 More Complex Port-Based VLAN


FSX
IP Subnet1
IPX Net 1
Atalk 100.1
Zone "A"
Port 17
IP Subnet2
IPX Net 2
Atalk 200.1
Zone "B"
Port 18
IP Subnet3
IPX Net 3
Atalk 300.1
Zone "C"
Port 19
IP Subnet4
IPX Net 4
Atalk 400.1
Zone "D"
Port 20
⊘ = STP Blocked VLAN
FESX-A
VLAN 2 Port 1-4 IP Sub1 IPXnet1 AT 100 Zone A
VLAN 3 Port 5-8 IP Sub2 IPXnet2 AT 200 Zone B
VLAN 4 Port 9-12 IP Sub3 IPXnet3 AT 300 Zone C
VLAN 5 Port 13-16 IP Sub4 IPXnet4 AT 400 Zone D
ROOT BRIDGE FOR VLAN - BLUE VLAN - RED
ROOT BRIDGE FOR VLAN - BROWN VLAN - GREEN
FESX-B
VLAN 2 Port 1-4 IP Sub1 IPXnet1 AT 100 Zone A
VLAN 3 Port 5-8 IP Sub2 IPXnet2 AT 200 Zone B
VLAN 4 Port 9-12 IP Sub3 IPXnet3 AT 300 Zone C
VLAN 5 Port 13-16 IP Sub4 IPXnet4 AT 400 Zone D
FESX-C
VLAN 2 Port 1-4 IP Sub1 IPXnet1 AT 100 Zone A
VLAN 3 Port 5-8 IP Sub2 IPXnet2 AT 200 Zone B
VLAN 4 Port 9-12 IP Sub3 IPXnet3 AT 300 Zone C
VLAN 5 Port 13-16 IP Sub4 IPXnet4 AT 400 Zone D

To configure the Port-based VLANs on the FESX Layer 2 Switches in Figure 14.10, use the following method.

Configuring FESX-A

Enter the following commands to configure FESX-A:

```
FastIron> enable
FastIron#configure terminal
FastIron(config)#hostname FESX-A
FastIron-A(config)#vlan 2 name BROWN
FastIron-A(config-vlan-2)#untag ethernet 1 to 4 ethernet 17
FastIron-A(config-vlan-2)#tag ethernet 25 to 26
FastIron-A(config-vlan-2)#spanning-tree
FastIron-A(config-vlan-2)#vlan 3 name GREEN
FastIron-A(config-vlan-3)#untag ethernet 5 to 8 ethernet 18
FastIron-A(config-vlan-3)#tag ethernet 25 to 26
FastIron-A(config-vlan-3)#spanning-tree
FastIron-A(config-vlan-3)#vlan 4 name BLUE
FastIron-A(config-vlan-4)#untag ethernet 9 to 12 ethernet 19
FastIron-A(config-vlan-4)#tag ethernet 25 to 26
FastIron-A(config-vlan-4)#spanning-tree
FastIron-A(config-vlan-4)#spanning-tree priority 500
FastIron-A(config-vlan-4)#vlan 5 name RED
FastIron-A(config-vlan-5)#untag ethernet 13 to 16 ethernet 20
FastIron-A(config-vlan-5)#tag ethernet 25 to 26
FastIron-A(config-vlan-5)#spanning-tree
FastIron-A(config-vlan-5)#spanning-tree priority 500
FastIron-A(config-vlan-5)#end
FastIron-A#write memory
```

Configuring FESX-B

Enter the following commands to configure FESX-B:

```
FastIron> en
FastIron#configure terminal
FastIron(config)#hostname FESX-B
FastIron-B(config)#vlan 2 name BROWN
FastIron-B(config-vlan-2)#untag ethernet 1 to 4
FastIron-B(config-vlan-2)#tag ethernet 25 to 26
FastIron-B(config-vlan-2)#spanning-tree
FastIron-B(config-vlan-2)#spanning-tree priority 500
FastIron-B(config-vlan-2)#vlan 3 name GREEN
FastIron-B(config-vlan-3)#untag ethernet 5 to 8
FastIron-B(config-vlan-3)#tag ethernet 25 to 26
FastIron-B(config-vlan-3)#spanning-tree
FastIron-B(config-vlan-3)#spanning-tree priority 500
FastIron-B(config-vlan-3)#vlan 4 name BLUE
FastIron-B(config-vlan-4)#untag ethernet 9 to 12
FastIron-B(config-vlan-4)#tag ethernet 25 to 26
FastIron-B(config-vlan-4)#vlan 5 name RED
FastIron-B(config-vlan-5)#untag ethernet 13 to 16
FastIron-B(config-vlan-5)#tag ethernet 25 to 26
FastIron-B(config-vlan-5)#end
FastIron-B#write memory
```

Configuring FESX-C

Enter the following commands to configure FESX-C:

```
FastIron> en
FastIron#configure terminal
FastIron(config)#hostname FESX-C
FastIron-C(config)#vlan 2 name BROWN
FastIron-C(config-vlan-2)#untag ethernet 1 to 4
FastIron-C(config-vlan-2)#tag ethernet 25 to 26
FastIron-C(config-vlan-2)#vlan 3 name GREEN
FastIron-C(config-vlan-3)#untag ethernet 5 to 8
FastIron-C(config-vlan-3)#tag ethernet 25 to 26
FastIron-C(config-vlan-3)#vlan 4 name BLUE
FastIron-C(config-vlan-4)#untag ethernet 9 to 12
FastIron-C(config-vlan-4)#tag ethernet 25 to 26
FastIron-C(config-vlan-4)#vlan 5 name RED
FastIron-C(config-vlan-5)#untag ethernet 13 to 16
FastIron-C(config-vlan-5)#tag ethernet 25 to 26
FastIron-C(config-vlan-5)#end
FastIron-C#write memory
```

Syntax: vlan <vlan-id> by port

Syntax: untagged ethernet [<slotnum>/]<portnum> [to [<slotnum>/]<portnum> | ethernet [<slotnum>/]<portnum>]

Syntax: tagged ethernet [<slotnum>/]<portnum> [to <[<slotnum>/]portnum> | ethernet [<slotnum>/]<portnum>]

Syntax: [no] spanning-tree

Syntax: spanning-tree [ethernet [<slotnum>/]<portnum> path-cost <value> priority <value>] forward-delay <value> hello-time <value> maximum-age <time> priority <value>

Modifying a Port-Based VLAN

You can make the following modifications to a port-based VLAN:

- Add or delete a VLAN port.

- Enable or disable STP.

Removing a Port-Based VLAN

Suppose you want to remove VLAN 5 from the example in Figure 14.10. To do so, use the following procedure.

1. Access the global CONFIG level of the CLI on FESX-A by entering the following commands:

```
FastIron-A> enable
No password has been assigned yet...
FastIron-A#configure terminal
FastIron-A(config)#
```

2. Enter the following command:

```
FastIron-A(config)#no vlan 5
FastIron-A(config)#
```

3. Enter the following commands to exit the CONFIG level and save the configuration to the system-config file on flash memory:

```
FastIron-A(config)#
FastIron-A(config)#end
FastIron-A#write memory
FastIron-A#
```

4. Repeat steps 1 – 3 on FESX-B.

Syntax: no vlan <vlan-id> by port

Removing a Port from a VLAN

Suppose you want to remove port 11 from VLAN 4 on FESX-A shown in Figure 14.10. To do so, use the following procedure.

1. Access the global CONFIG level of the CLI on FESX-A by entering the following command:

```
FastIron-A> enable
No password has been assigned yet...
FastIron-A#configure terminal
FastIron-A(config)#
```

2. Access the level of the CLI for configuring port-based VLAN 4 by entering the following command:

```
FastIron-A(config)#
FastIron-A(config)#vlan 4
FastIron-A(config-vlan-4)#
```

3. Enter the following commands:

```
FastIron-A(config-vlan-4)#
FastIron-A(config-vlan-4)#no untag ethernet 11
deleted port ethe 11 from port-vlan 4.
FastIron-A(config-vlan-4)#
```

4. Enter the following commands to exit the VLAN CONFIG mode and save the configuration to the system-config file on flash memory:

```
FastIron-A(config-vlan-4)#
FastIron-A(config-vlan-4)#end
FastIron-A#write memory
```

You can remove all the ports from a port-based VLAN without losing the rest of the VLAN’s configuration. However, you cannot configure an IP address on a virtual routing interface unless the VLAN contains ports. If the VLAN has a virtual routing interface, the virtual routing interface’s IP address is deleted when the ports associated with the interface are deleted. The rest of the VLAN configuration is retained.

Enable Spanning Tree on a VLAN

The spanning tree bridge and port parameters are configurable using one CLI command set at the Global Configuration Level of each Port-based VLAN. Suppose you want to enable the IEEE 802.1D STP across VLAN 3. To do so, use the following method.

NOTE: When port-based VLANs are not operating on the system, STP is set on a system-wide level at the global CONFIG level of the CLI.

---

1. Access the global CONFIG level of the CLI on FESX-A by entering the following commands:

```
FastIron-A> enable
No password has been assigned yet...
FastIron-A#configure terminal
FastIron-A(config)#
```

2. Access the level of the CLI for configuring port-based VLAN 3 by entering the following command:

```
FastIron-A(config)#
FastIron-A(config)#vlan 3
FastIron-A(config-vlan-3)#
```

3. From VLAN 3's configuration level of the CLI, enter the following command to enable STP on all tagged and untagged ports associated with VLAN 3.

```
FastIron-B(config-vlan-3)#
FastIron-B(config-vlan-3)#spanning-tree
FastIron-B(config-vlan-3)#
```

4. Enter the following commands to exit the VLAN CONFIG mode and save the configuration to the system-config file on flash memory:

```
FastIron-B(config-vlan-3)#
FastIron-B(config-vlan-3)#end
FastIron-B#write memory
FastIron-B#
```

5. Repeat steps 1 – 4 on FESX-B.

NOTE: You do not need to configure values for the STP parameters. All parameters have default values as noted below. Additionally, all values will be globally applied to all ports on the system or on the port-based VLAN for which they are defined.

To configure a specific path-cost or priority value for a given port, enter those values using the key words in the brackets [ ] shown in the syntax summary below. If you do not want to specify values for any given port, this portion of the command is not required.

Syntax: vlan <vlan-id> by port

Syntax: [no] spanning-tree

Syntax: spanning-tree [ethernet [<slotnum>/]<portnum> path-cost <value> priority <value>] forward-delay <value> hello-time <value> maximum-age <time> priority <value>

Bridge STP Parameters (applied to all ports within a VLAN)

- Forward Delay – the period of time a bridge will wait (the listen and learn period) before forwarding data packets. Possible values: 4 – 30 seconds. Default is 15.
- Maximum Age – the interval a bridge will wait for receipt of a hello packet before initiating a topology change. Possible values: 6 – 40 seconds. Default is 20.

- Hello Time – the interval of time between each configuration BPDU sent by the root bridge. Possible values: 1 – 10 seconds. Default is 2.
- Priority – a parameter used to identify the root bridge in a network. The bridge with the lowest value has the highest priority and is the root. Possible values: 1 – 65,535. Default is 32,678.

Port Parameters (applied to a specified port within a VLAN)

- Path Cost – a parameter used to assign a higher or lower path cost to a port. Possible values: 1 – 65535. Default is (1000/Port Speed) for Half-Duplex ports and is (1000/Port Speed)/2 for Full-Duplex ports.
- Priority – value determines when a port will be rerouted in relation to other ports. Possible values: 0 – 255. Default is 128.

Configuring IP Subnet, IPX Network and Protocol-Based VLANs

Protocol-based VLANs provide the ability to define separate broadcast domains for several unique Layer 3 protocols within a single Layer 2 broadcast domain. Some applications for this feature might include security between departments with unique protocol requirements. This feature enables you to limit the amount of broadcast traffic end-stations, servers, and routers need to accept.

Configuration Example

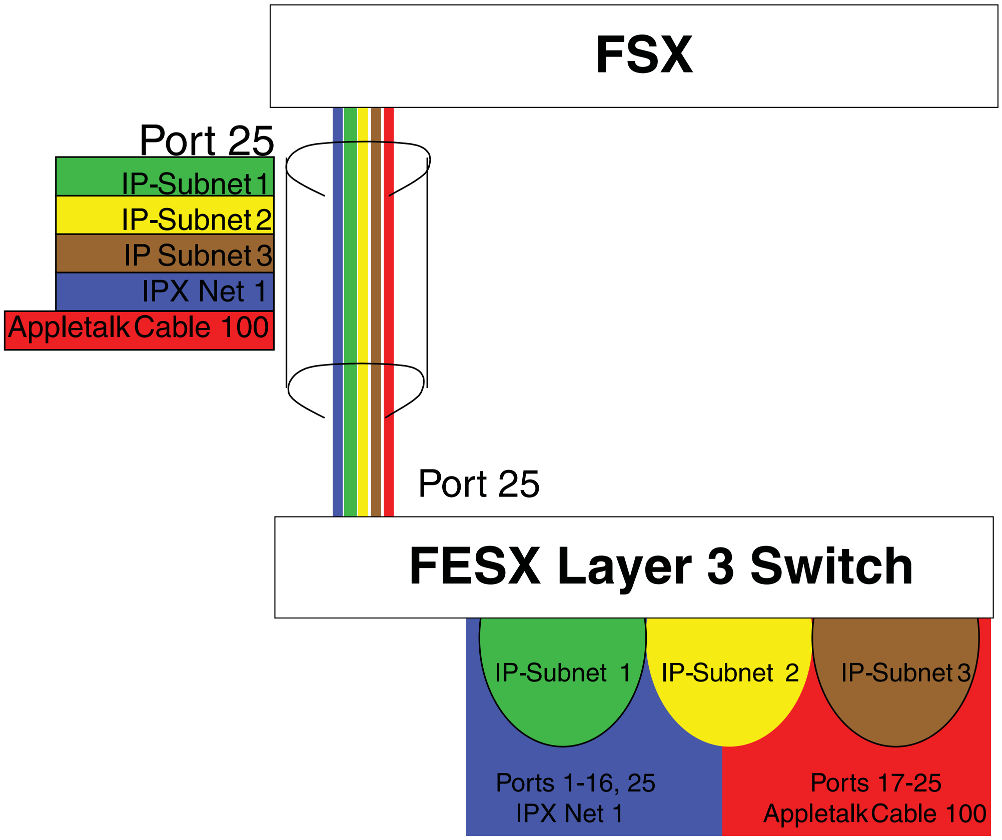
Suppose you want to create five separate Layer 3 broadcast domains within a single Layer 2 STP broadcast domain:

- Three broadcast domains, one for each of three separate IP subnets
- One for IPX Network 1
- One for the Appletalk protocol

Also suppose you want a single router interface to be present within all of these separate broadcast domains, without using IEEE 802.1Q VLAN tagging or any proprietary form of VLAN tagging.

Figure 14.11 shows this configuration.

Figure 14.11 Protocol-Based (Layer 3) VLANs



To configure the VLANs shown in Figure 14.11, use the following procedure.

1. To permanently assign ports 1 – 8 and port 25 to IP subnet VLAN 1.1.1.0, enter the following commands:

```
FastIron> en
No password has been assigned yet...
FastIron#config t
FastIron(config)#
FastIron(config)#ip-subnet 1.1.1.0/24 name Green
FastIron(config-ip-subnet)#no dynamic
FastIron(config-ip-subnet)#static ethernet 1 to 8 ethernet 25
```

2. To permanently assign ports 9 – 16 and port 25 to IP subnet VLAN 1.1.2.0, enter the following commands:

```
FastIron(config-ip-subnet)#ip-subnet 1.1.2.0/24 name Yellow
FastIron(config-ip-subnet)#no dynamic
FastIron(config-ip-subnet)#static ethernet 9 to 16 ethernet 25
```

3. To permanently assign ports 17 – 25 to IP subnet VLAN 1.1.3.0, enter the following commands:

```
FastIron(config-ip-subnet)#ip-subnet 1.1.3.0/24 name Brown
FastIron(config-ip-subnet)#no dynamic
FastIron(config-ip-subnet)#static ethernet 17 to 25
```

4. To permanently assign ports 1 – 12 and port 25 to IPX network 1 VLAN, enter the following commands:

```
FastIron(config-ip-subnet)#ipx-network 1 ethernet_802.3 name Blue
FastIron(config-ipx-network)#no dynamic
FastIron(config-ipx-network)#static ethernet 1 to 12 ethernet 25
FastIron(config-ipx-network)#
```

5. To permanently assign ports 12 – 25 to Appletalk VLAN, enter the following commands:

```
FastIron(config-ipx-proto)#atalk-proto name Red
FastIron(config-atalk-proto)#no dynamic
FastIron(config-atalk-proto)#static ethernet 13 to 25
FastIron(config-atalk-proto)#end
FastIron#write memory
FastIron#
```

Syntax: ip-subnet <ip-addr> <ip-mask> [name <string>]

Syntax: ipx-network <ipx-network-number> <frame-encapsulation-type> netbios-allow | netbios-disallow [name <string>]

Syntax: ip-proto | ipx-proto | atalk-proto | decnet-proto | netbios-proto | other-proto static | exclude | dynamic ethernet [<slotnum>/]<portnum> [to [<slotnum>/]<portnum>] [name <string>]

Configuring IP Subnet, IPX Network, and Protocol-Based VLANs Within Port-Based VLANs

If you plan to use port-based VLANs in conjunction with protocol-based VLANs, you must create the port-based VLANs first. Once you create a port-based VLAN, then you can assign Layer 3 protocol VLANs within the boundaries of the port-based VLAN. Generally, you create port-based VLANs to allow multiple separate STP domains.

EXAMPLES:

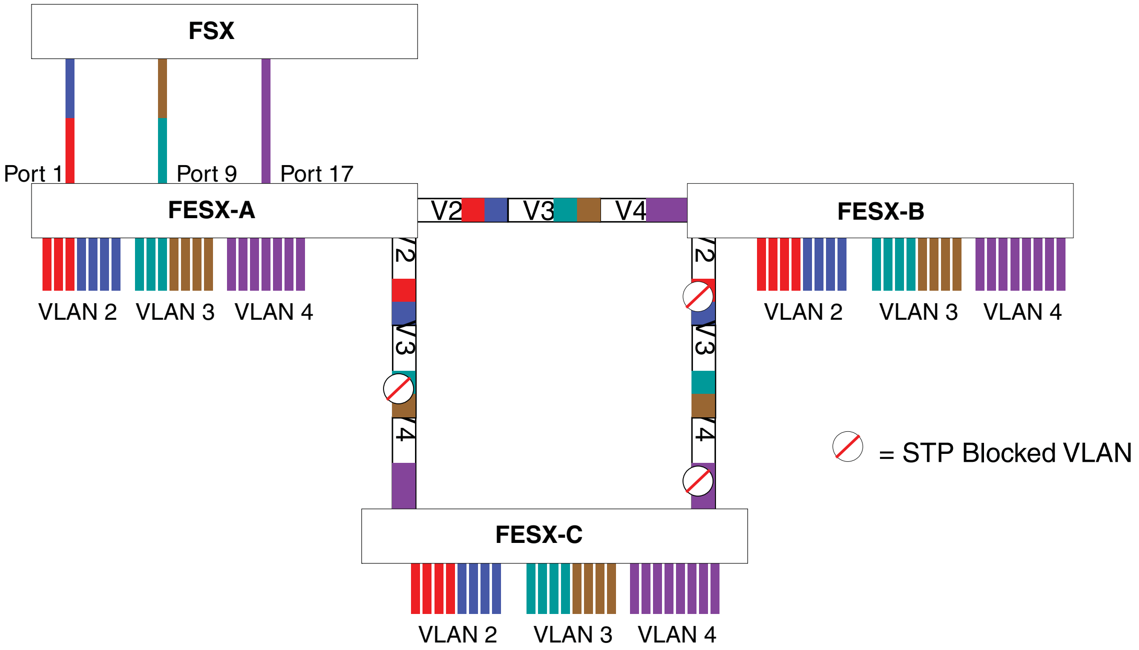
Suppose you need to provide three separate STP domains across an enterprise campus backbone. The first STP domain (VLAN 2) requires a set of ports at each Layer 2 Switch location to be statically mapped to IP only. No other protocols can enter the switches on this set of ports.

A second set of ports within STP domain VLAN 2 will be restricted to only IPX traffic. The IP and IPX protocol VLANs will overlap on Port 1 of FESX-A to support both protocols on the same router interface. The IP subnets and IPX network that span the two protocol VLANs will be determined by the FastIron router configuration. The IP and IPX Protocol VLANs ensure that only the ports included in the each Layer 3 protocol VLAN will see traffic from the FastIron router.

The second STP domain (VLAN 3) requires that half the ports in the domain are dedicated to IP subnet 1.1.1.0/24 and the other ports are dedicated to IPX network 1. Similar to VLAN 2, Port 9 from VLAN 3 will be used to carry this IP subnet and IPX network to the FastIron router. No other protocols will be allowed to enter the network on VLAN 3. Also, no IP packets with a source address on subnet 1.1.1.0/24 or IPX packets with a source address on network 1 will be allowed to enter the switches on VLAN 3.

There is no need to segment Layer 3 broadcast domains within the STP broadcast domain (VLAN 4). The FastIron router will dictate the IP subnets and IPX network that are on VLAN 4. There are no Layer 3 protocol restrictions on VLAN 4; however, the FastIron router is configured to only forward IP and IPX between STP domains.

Figure 14.12 More Protocol-Based VLANs



To configure the Layer 3 VLANs on the FESX Layer 2 Switches in Figure 14.12, use the following procedure.

Configuring FESX-A

Enter the following commands to configure FESX-A:

1. Create port-based VLAN 2 and assign the untagged and tagged ports that will participate in this VLAN:

```
FastIron-A >en
FastIron-A#config t
FastIron-A(config)#vlan 2 name IP_IPX_Protocol
FastIron-A(config-vlan-2)#untag e1 to 8
FastIron-A(config-vlan-2)#tag e25 to 26
```

2. Enable STP and set the priority to force FESX-A to be the root bridge for VLAN 2:

```
FastIron-A(config-vlan-2)#spanning-tree
FastIron-A(config-vlan-2)#spanning-tree priority 500
FastIron-A(config-vlan-2)#
```

3. Create the IP and IPX protocol-based VLANs and statically assign the ports within VLAN 2 that will be associated with each protocol-based VLAN:

```
FastIron-A(config-vlan-2)#ip-proto name Red
FastIron-A(config-vlan-ip-proto)#no dynamic
FastIron-A(config-vlan-ip-proto)#static e1 to 4 e25 to 26
FastIron-A(config-vlan-ip-proto)#exclude e5 to 8
FastIron-A(config-vlan-ip-proto)#ipx-proto name Blue
FastIron-A(config-vlan-ipx-proto)#no dynamic
FastIron-A(config-vlan-ipx-proto)#static e1 e5 to 8 e25 to 26
FastIron-A(config-vlan-ipx-proto)#exclude e2 to 4
```

4. To prevent machines with non-IP protocols from getting into the IP portion of VLAN 2, create another Layer 3 protocol VLAN to exclude all other protocols from the ports that contains the IP-protocol VLAN. To do so, enter the following commands:

```
FastIron-A(config-vlan-ipx-proto)#other-proto name Block_other_proto
FastIron-A(config-vlan-other-proto)#no dynamic
```

```
FastIron-A(config-vlan-other-proto)#exclude e1 to 8
FastIron-A(config-vlan-other-proto)#
```

5. Create port-based VLAN 3. Note that FESX-B will be the root for this STP domain, so you do not need to adjust the STP priority.

```
FastIron-A(config-vlan-other-proto)#vlan 3 name IP-Sub_IPX-Net_Vlans
FastIron-A(config-vlan-3)#untag e9 to 16
FastIron-A(config-vlan-3)#tag e25 to 26
FastIron-A(config-vlan-3)#spanning-tree
FastIron-A(config-vlan-3)#
```

6. Create IP subnet VLAN 1.1.1.0/24, IPX network 1, and other-protocol VLANs

```
FastIron-A(config-vlan-3)#ip-subnet 1.1.1.0/24 name Green
FastIron-A(config-vlan-ip-subnet)#no dynamic
FastIron-A(config-vlan-ip-subnet)#static e9 to 12 e25 to 26
FastIron-A(config-vlan-ip-subnet)#exclude e13 to 16
FastIron-A(config-vlan-ip-subnet)#ipx-net 1 ethernet_802.3 name Brown
FastIron-A(config-vlan-ipx-network)#no dynamic
FastIron-A(config-vlan-ipx-network)#static e9 e13 to 16 e25 to 26
FastIron-A(config-vlan-ipx-network)#exclude e10 to 12
FastIron-A(config-vlan-ipx-network)#other-proto name Block_other_proto
FastIron-A(config-vlan-other-proto)#no dynamic
FastIron-A(config-vlan-other-proto)#exclude e9 to 16
FastIron-A(config-vlan-other-proto)#
```

7. Configure the last port-based VLAN 4. You need to set the STP priority for this VLAN because FESX-A will be the root bridge for this VLAN. Since you do not need to partition this STP domain into multiple Layer 3 broadcast domains, this is the only configuration required for VLAN 4:

```
FastIron-A(config-vlan-other-proto)#vlan 4 name Purple_ALL-Protocols
FastIron-A(config-vlan-4)#untag e17 to 24
FastIron-A(config-vlan-4)#tag e25 to 26
FastIron-A(config-vlan-4)#spanning-tree
FastIron-A(config-vlan-4)#spanning-tree priority 500
FastIron-A(config-vlan-4)#
```

Configuring FESX-B

Enter the following commands to configure FESX-B:

```
FastIron#config t
FastIron(config)#host FastIron-B
FastIron-B(config)#vlan 2 name IP_IPX_Protocol
FastIron-B(config-vlan-2)#untag e1 to 8
FastIron-B(config-vlan-2)#tag e25 to 26
FastIron-B(config-vlan-2)#spanning-tree
FastIron-B(config-vlan-2)#ip-proto name Red
FastIron-B(config-vlan-ip-proto)#no dynamic
FastIron-B(config-vlan-ip-proto)#static e1 to 4 e25 to 26
FastIron-B(config-vlan-ip-proto)#exclude e5 to 8
FastIron-B(config-vlan-ip-proto)#ipx-proto name Blue
FastIron-B(config-vlan-ipx-proto)#no dynamic
FastIron-B(config-vlan-ipx-proto)#static e5 to 8 e25 to 26
FastIron-B(config-vlan-ipx-proto)#exclude e1 to 4
FastIron-B(config-vlan-other-proto)#vlan 3 name IP-Sub_IPX-Net_VLANs
FastIron-B(config-vlan-3)#untag e9 to 16
FastIron-B(config-vlan-3)#tag e25 to 26
FastIron-B(config-vlan-3)#spanning-tree
FastIron-B(config-vlan-3)#spanning-tree priority 500
FastIron-B(config-vlan-3)#ip-sub 1.1.1.0/24 name Green
FastIron-B(config-vlan-ip-subnet)#no dynamic
```

```
FastIron-B(config-vlan-ip-subnet)#static e9 to 12 e25 to 26
FastIron-B(config-vlan-ip-subnet)#exclude e13 to 16
FastIron-B(config-vlan-ip-subnet)#ipx-net 1 ethernet_802.3 name Brown
FastIron-B(config-vlan-ipx-network)#no dynamic
FastIron-B(config-vlan-ipx-network)#static e13 to 16 e25 to 26
FastIron-B(config-vlan-ipx-network)#exclude e9 to 12
FastIron-B(config-vlan-ipx-network)#vlan 4 name Purple_ALL-Protocols
FastIron-B(config-vlan-4)#untag e17 to 24
FastIron-B(config-vlan-4)#tag e25 to 26
FastIron-B(config-vlan-4)#spanning-tree
```

Configuring FESX-C

Enter the following commands to configure FESX-C:

```
FastIron#config t
FastIron(config)#host FastIron-C
FastIron-C(config)#vlan 2 name IP_IPX_Protocol
FastIron-C(config-vlan-2)#untag e1 to 8
FastIron-C(config-vlan-2)#tag e25 to 26
FastIron-C(config-vlan-2)#spanning-tree
FastIron-C(config-vlan-2)#ip-proto name Red
FastIron-C(config-vlan-ip-proto)#no dynamic
FastIron-C(config-vlan-ip-proto)#static e1 to 4 e25 to 26
FastIron-C(config-vlan-ip-proto)#exclude e5 to 8
FastIron-C(config-vlan-ip-proto)#ipx-proto name Blue
FastIron-C(config-vlan-ipx-proto)#no dynamic
FastIron-C(config-vlan-ipx-proto)#static e5 to 8 e25 to 26
FastIron-C(config-vlan-ipx-proto)#exclude e1 to 4
FastIron-C(config-vlan-other-proto)#vlan 3 name IP-Sub_IPX-Net_VLANs
FastIron-C(config-vlan-3)#untag e9 to 16
FastIron-C(config-vlan-3)#tag e25 to 26
FastIron-C(config-vlan-3)#spanning-tree
FastIron-C(config-vlan-3)#ip-sub 1.1.1.0/24 name Green
FastIron-C(config-vlan-ip-subnet)#no dynamic
FastIron-C(config-vlan-ip-subnet)#static e9 to 12 e25 to 26
FastIron-C(config-vlan-ip-subnet)#exclude e13 to 16
FastIron-C(config-vlan-ip-subnet)#ipx-net 1 ethernet_802.3 name Brown
FastIron-C(config-vlan-ipx-network)#no dynamic
FastIron-C(config-vlan-ipx-network)#static e13 to 16 e25 to 26
FastIron-C(config-vlan-ipx-network)#exclude e9 to 12
FastIron-C(config-vlan-ipx-network)#vlan 4 name Purple_ALL-Protocols
FastIron-C(config-vlan-4)#untag e17 to 24
FastIron-C(config-vlan-4)#tag e25 to 26
FastIron-C(config-vlan-4)#spanning-tree
```

Configuring an IPv6 Protocol VLAN

You can configure a protocol-based VLAN as a broadcast domain for IPv6 traffic. When the Layer 3 Switch receives an IPv6 multicast packet (a packet with 06 in the version field and 0xFF as the beginning of the destination address), the Layer 3 Switch forwards the packet to all other ports in the VLAN.

NOTE: The Layer 3 Switch forwards all IPv6 multicast packets to all ports in the VLAN except the port that received the packet, and does not distinguish among subnet directed multicasts.

You can add the VLAN ports as static ports or dynamic ports. A static port is always an active member of the VLAN. Dynamic ports within any protocol VLAN age out after 10 minutes if no member protocol traffic is received on a port within the VLAN. The aged out port, however, remains as a candidate dynamic port for that VLAN. The port becomes active in the VLAN again if member protocol traffic is received on that port.

Once a port is re-activated, the aging out period for the port is reset to 20 minutes. Each time a member protocol packet is received by a candidate dynamic port (aged out port) the port becomes active again and the aging out period is reset for 20 minutes.

NOTE: Starting in software release 02.5.00, you can disable VLAN membership aging of dynamically added ports. See “Disabling Membership Aging of Dynamic VLAN Ports” on page 14-34**)**.

To configure an IPv6 VLAN, enter commands such as the following:

```
FastIron(config)#vlan 2
FastIron(config-vlan-2)#untag ethernet 1/1 to 1/8
FastIron(config-vlan-2)#ipv6-proto name V6
FastIron(config-ipv6-subnet)#static ethernet 1/1 to 1/6
FastIron(config-ipv6-subnet)#dynamic
```

The first two commands configure a port-based VLAN and add ports 1/1 – 1/8 to the VLAN. The remaining commands configure an IPv6 VLAN within the port-based VLAN. The **static** command adds ports 1/1 – 1/6 as static ports, which do not age out. The **dynamic** command adds the remaining ports, 1/7 – 1/8, as dynamic ports. These ports are subject to aging as described above.

Syntax: [no] ipv6-proto [name <string>]

Routing Between VLANs Using Virtual Routing Interfaces (Layer 3 Switches Only)

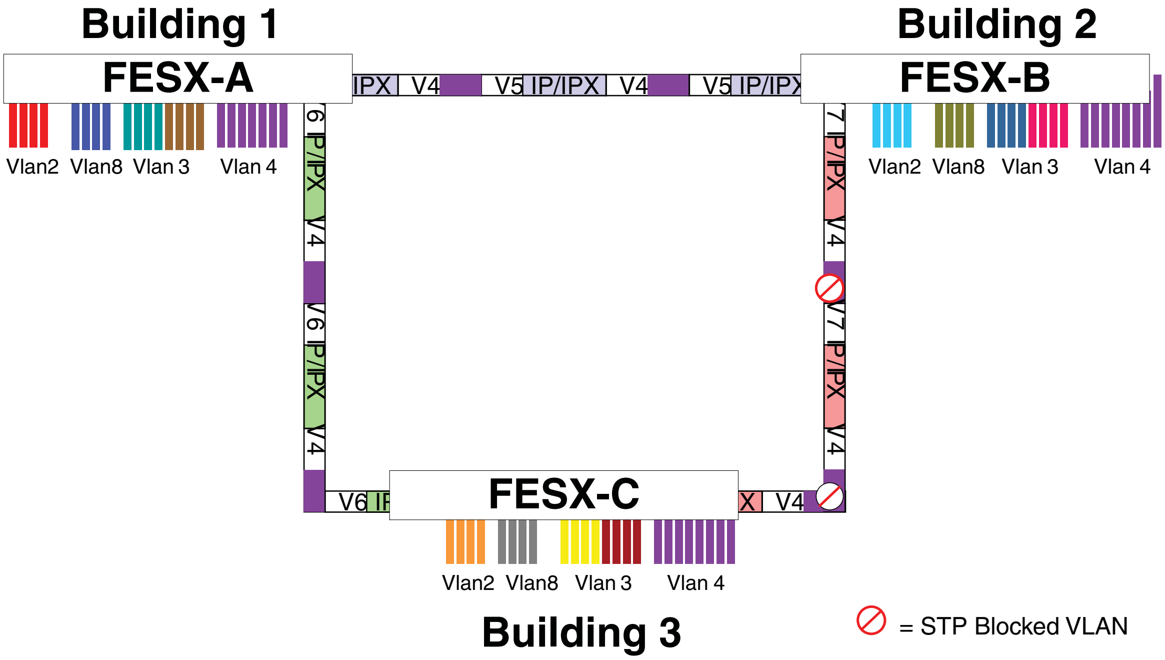
Foundry Layer 3 Switches offer the ability to create a virtual routing interface within a Layer 2 STP port-based VLAN or within each Layer 3 protocol, IP subnet, or IPX network VLAN. This combination of multiple Layer 2 and/ or Layer 3 broadcast domains and virtual routing interfaces are the basis for Foundry Networks’ very powerful Integrated Switch Routing (ISR) technology. ISR is very flexible and can solve many networking problems. The following example is meant to provide ideas by demonstrating some of the concepts of ISR.

Example: Suppose you want to move routing out to each of three buildings in a network. Remember that the only protocols present on VLAN 2 and VLAN 3 are IP and IPX. Therefore, you can eliminate tagged ports 25 and 26 from both VLAN 2 and VLAN 3 and create new tagged port-based VLANs to support separate IP subnets and IPX networks for each backbone link.

You also need to create unique IP subnets and IPX networks within VLAN 2 and VLAN 3 at each building. This will create a fully routed IP and IPX backbone for VLAN 2 and VLAN 3. However, VLAN 4 has no protocol restrictions across the backbone. In fact there are requirements for NetBIOS and DecNet to be bridged among the three building locations. The IP subnet and IPX network that exists within VLAN 4 must remain a flat Layer 2 switched STP domain. You enable routing for IP and IPX on a virtual routing interface only on FESX-A. This will provide the flat IP and IPX segment with connectivity to the rest of the network. Within VLAN 4 IP and IPX will follow the STP topology. All other IP subnets and IPX networks will be fully routed and have use of all paths at all times during normal operation.

Figure 14.13 shows the configuration described above.

Figure 14.13 Routing Between Protocol-Based VLANs


Building 1
FESX-A
Vlan2 Vlan8 Vlan 3 Vlan 4
Building 2
FESX-B
Vlan2 Vlan8 Vlan 3 Vlan 4
FESX-C
Vlan2 Vlan8 Vlan 3 Vlan 4
Building 3
IPX V4 V5 IP/IPX V4 V5 IP/IPX
V6 IP/IPX V4 V6 IP/IPX V4
V7 IP/IPX V4 V7 IP/IPX V4
V6 IP
X V4
= STP Blocked VLAN

To configure the Layer 3 VLANs and virtual routing interfaces on the FESX Layer 3 Switch in Figure 14.13, use the following procedure.

Configuring FESX-A

Enter the following commands to configure FESX-A. The following commands enable OSPF or RIP routing.

```
FastIron> en
No password has been assigned yet...
FastIron#configure terminal
FastIron(config)#hostname FESX-A
FastIron-A(config)#router ospf
FastIron-A(config-ospf-router)#area 0.0.0.0 normal
Please save configuration to flash and reboot.
FastIron-A(config-ospf-router)#
```

The following commands create the port-based VLAN 2. In the previous example, an external FESX defined the router interfaces for VLAN 2. With ISR, routing for VLAN 2 is done locally within each FESX. Therefore, there are two ways you can solve this problem. One way is to create a unique IP subnet and IPX network VLAN, each with its own virtual routing interface and unique IP or IPX address within VLAN 2 on each FESX. In this example, this is the configuration used for VLAN 3. The second way is to split VLAN 2 into two separate port-based VLANs and create a virtual router interface within each port-based VLAN. Later in this example, this second option is used to create a port-based VLAN 8 to show that there are multiple ways to accomplish the same task with ISR.

You also need to create the Other-Protocol VLAN within port-based VLAN 2 and 8 to prevent unwanted protocols from being Layer 2 switched within port-based VLAN 2 or 8. Note that the only port-based VLAN that requires STP in this example is VLAN 4. You will need to configure the rest of the network to prevent the need to run STP.

```
FastIron-A(config-ospf-router)#vlan 2 name IP-Subnet_1.1.2.0/24
FastIron-A(config-vlan-2)#untag e 1 to 4
FastIron-A(config-vlan-2)#no spanning-tree
FastIron-A(config-vlan-2)#router-interface ve1
FastIron-A(config-vlan-2)#other-proto name block_other_protocols
FastIron-A(config-vlan-other-proto)#no dynamic
FastIron-A(config-vlan-other-proto)#exclude e 1 to 4
```

Once you have defined the port-based VLAN and created the virtual routing interface, you need to configure the virtual routing interface just as you would configure a physical interface.

```
FastIron-A(config-vlan-other-proto)#interface ve1
FastIron-A(config-vif-1)#ip address 1.1.2.1/24
FastIron-A(config-vif-1)#ip ospf area 0.0.0.0
```

Do the same thing for VLAN 8.

```
FastIron-A(config-vif-1)#vlan 8 name IPX_Network2
FastIron-A(config-vlan-8)#untag ethernet 5 to 8
FastIron-A(config-vlan-8)#no spanning-tree
FastIron-A(config-vlan-8)#router-interface ve 2
FastIron-A(config-vlan-8)#other-proto name block-other-protocols
FastIron-A(config-vlan-other-proto)#no dynamic
FastIron-A(config-vlan-other-proto)#exclude ethernet 5 to 8
FastIron-A(config-vlan-other-proto)#int ve2
FastIron-A(config-vif-2)#ipx network 2 ethernet_802.3
FastIron-A(config-vif-2)#
```

The next thing you need to do is create VLAN 3. This is very similar to the previous example with the addition of virtual routing interfaces to the IP subnet and IPX network VLANs. Also there is no need to exclude ports from the IP subnet and IPX network VLANs on the router.

```
FastIron-A(config-vif-2)#vlan 3 name IP_Sub_&_IPX_Net_VLAN
FastIron-A(config-vlan-3)#untag e 9 to 16
FastIron-A(config-vlan-3)#no spanning-tree
FastIron-A(config-vlan-3)#ip-subnet 1.1.1.0/24
FastIron-A(config-vlan-ip-subnet)#static e 9 to 12
FastIron-A(config-vlan-ip-subnet)#router-interface ve3
FastIron-A(config-vlan-ip-subnet)#ipx-network 1 ethernet_802.3
FastIron-A(config-vlan-ipx-network)#static e 13 to 16
FastIron-A(config-vlan-ipx-network)#router-interface ve4
FastIron-A(config-vlan-ipx-network)#other-proto name block-other-protocols
FastIron-A(config-vlan-other-proto)#exclude e 9 to 16
FastIron-A(config-vlan-other-proto)#no dynamic
FastIron-A(config-vlan-other-proto)#interface ve 3
FastIron-A(config-vif-3)#ip addr 1.1.1.1/24
FastIron-A(config-vif-3)#ip ospf area 0.0.0.0
FastIron-A(config-vif-3)#int ve4
FastIron-A(config-vif-4)#ipx network 1 ethernet_802.3
FastIron-A(config-vif-4)#
```

Now configure VLAN 4. Remember this is a flat segment that, in the previous example, obtained its IP default gateway and IPX router services from an external FESX. In this example, FESX-A will provide the routing services for VLAN 4. You also want to configure the STP priority for VLAN 4 to make FESX-A the root bridge for this VLAN.

```
FastIron-A(config-vif-4)#vlan 4 name Bridged_ALL_Protocols
FastIron-A(config-vlan-4)#untag ethernet 17 to 24
FastIron-A(config-vlan-4)#tag ethernet 25 to 26
FastIron-A(config-vlan-4)#spanning-tree
FastIron-A(config-vlan-4)#spanning-tree priority 500
FastIron-A(config-vlan-4)#router-interface ve5
FastIron-A(config-vlan-4)#int ve5
FastIron-A(config-vif-5)#ip address 1.1.3.1/24
FastIron-A(config-vif-5)#ip ospf area 0.0.0.0
FastIron-A(config-vif-5)#ipx network 3 ethernet_802.3
FastIron-A(config-vif-5)#
```

It is time to configure a separate port-based VLAN for each of the routed backbone ports (Ethernet 25 and 26). If you do not create a separate tagged port-based VLAN for each point-to-point backbone link, you need to include tagged interfaces for Ethernet 25 and 26 within VLANs 2, 3, and 8. This type of configuration makes the entire backbone a single STP domain for each VLAN 2, 3, and 8. This is the configuration used in the example in "Configuring IP Subnet, IPX Network and Protocol-Based VLANs" on page 14-22. In this scenario, the virtual routing interfaces within port-based VLANs 2, 3, and 8 will be accessible using only one path through the network. The path that is blocked by STP is not available to the routing protocols until it is in the STP FORWARDING state.

```
FastIron-A(config-vif-5)#vlan 5 name Rtr_BB_to_Bldg.2
FastIron-A(config-vlan-5)#tag e 25
FastIron-A(config-vlan-5)#no spanning-tree
FastIron-A(config-vlan-5)#router-interface ve6
FastIron-A(config-vlan-5)#vlan 6 name Rtr_BB_to_Bldg.3
FastIron-A(config-vlan-6)#tag ethernet 26
FastIron-A(config-vlan-6)#no spanning-tree
FastIron-A(config-vlan-6)#router-interface ve7
FastIron-A(config-vlan-6)#int ve6
FastIron-A(config-vif-6)#ip addr 1.1.4.1/24
FastIron-A(config-vif-6)#ip ospf area 0.0.0.0
FastIron-A(config-vif-6)#ipx network 4 ethernet_802.3
FastIron-A(config-vif-6)#int ve7
FastIron-A(config-vif-7)#ip addr 1.1.5.1/24
FastIron-A(config-vif-7)#ip ospf area 0.0.0.0
FastIron-A(config-vif-7)#ipx network 5 ethernet_802.3
FastIron-A(config-vif-7)#
```

This completes the configuration for FESX-A. The configuration for FESX-B and C is very similar except for a few issues.

- IP subnets and IPX networks configured on FESX-B and FESX-C must be unique across the entire network, except for the backbone port-based VLANs 5, 6, and 7 where the subnet is the same but the IP address must change.
- There is no need to change the default priority of STP within VLAN 4.
- There is no need to include a virtual router interface within VLAN 4.
- The backbone VLAN between FESX-B and FESX-C must be the same at both ends and requires a new VLAN ID. The VLAN ID for this port-based VLAN is VLAN 7.

Configuration for FESX-B

Enter the following commands to configure FESX-B.

```
FastIron> en
No password has been assigned yet...
FastIron#config t
FastIron(config)#hostname FESX-B
FastIron-B(config)#router ospf
FastIron-B(config-ospf-router)#area 0.0.0.0 normal
FastIron-B(config-ospf-router)#router ipx
FastIron-B(config-ospf-router)#vlan 2 name IP-Subnet_1.1.6.0/24
FastIron-B(config-vlan-2)#untag e 1 to 4
FastIron-B(config-vlan-2)#no spanning-tree
FastIron-B(config-vlan-2)#router-interface ve1
FastIron-B(config-vlan-2)#other-proto name block-other-protocols
FastIron-B(config-vlan-other-proto)#no dynamic
FastIron-B(config-vlan-other-proto)#exclude e 1 to 4
FastIron-B(config-vlan-other-proto)#int ve1
FastIron-B(config-vif-1)#ip addr 1.1.6.1/24
FastIron-B(config-vif-1)#ip ospf area 0.0.0.0
FastIron-B(config-vif-1)#vlan 8 name IPX_Network6
FastIron-B(config-vlan-8)#untag e 5 to 8
FastIron-B(config-vlan-8)#no span
FastIron-B(config-vlan-8)#router-int ve2
```

```
FastIron-B(config-vlan-8)#other-proto name block-other-protocols
FastIron-B(config-vlan-other-proto)#no dynamic
FastIron-B(config-vlan-other-proto)#exclude e 5 to 8
FastIron-B(config-vlan-other-proto)#int ve2
FastIron-B(config-vif-2)#ipx net 6 ethernet_802.3
FastIron-B(config-vif-2)#vlan 3 name IP_Sub_&_IPX_Net_VLAN
FastIron-B(config-vlan-3)#untag e 9 to 16
FastIron-B(config-vlan-3)#no spanning-tree
FastIron-B(config-vlan-3)#ip-subnet 1.1.7.0/24
FastIron-B(config-vlan-ip-subnet)#static e 9 to 12
FastIron-B(config-vlan-ip-subnet)#router-interface ve3
FastIron-B(config-vlan-ip-subnet)#ipx-network 7 ethernet_802.3
FastIron-B(config-vlan-ipx-network)#static e 13 to 16
FastIron-B(config-vlan-ipx-network)#router-interface ve4
FastIron-B(config-vlan-ipx-network)#other-proto name block-other-protocols
FastIron-B(config-vlan-other-proto)#exclude e 9 to 16
FastIron-B(config-vlan-other-proto)#no dynamic
FastIron-B(config-vlan-other-proto)#interface ve 3
FastIron-B(config-vif-3)#ip addr 1.1.7.1/24
FastIron-B(config-vif-3)#ip ospf area 0.0.0.0
FastIron-B(config-vif-3)#int ve4
FastIron-B(config-vif-4)#ipx network 7 ethernet_802.3
FastIron-B(config-vif-4)#vlan 4 name Bridged_ALL_Protocols
FastIron-B(config-vlan-4)#untag ethernet 17 to 24
FastIron-B(config-vlan-4)#tag ethernet 25 to 26
FastIron-B(config-vlan-4)#spanning-tree
FastIron-B(config-vlan-4)#vlan 5 name Rtr_BB_to_Bldg.1
FastIron-B(config-vlan-5)#tag e 25
FastIron-B(config-vlan-5)#no spanning-tree
FastIron-B(config-vlan-5)#router-interface ve5
FastIron-B(config-vlan-5)#vlan 7 name Rtr_BB_to_Bldg.3
FastIron-B(config-vlan-7)#tag ethernet 26
FastIron-B(config-vlan-7)#no spanning-tree
FastIron-B(config-vlan-7)#router-interface ve6
FastIron-B(config-vlan-7)#int ve5
FastIron-B(config-vif-5)#ip addr 1.1.4.2/24
FastIron-B(config-vif-5)#ip ospf area 0.0.0.0
FastIron-B(config-vif-5)#ipx network 4 ethernet_802.3
FastIron-B(config-vif-5)#int ve6
FastIron-B(config-vif-6)#ip addr 1.1.8.1/24
FastIron-B(config-vif-6)#ip ospf area 0.0.0.0
FastIron-B(config-vif-6)#ipx network 8 ethernet_802.3
FastIron-B(config-vif-6)#
```

Configuration for FESX-C

Enter the following commands to configure FESX-C.

```
FastIron> en
No password has been assigned yet...
FastIron#config t
FastIron(config)#hostname FESX-C
FastIron-C(config)#router ospf
FastIron-C(config-ospf-router)#area 0.0.0.0 normal
FastIron-C(config-ospf-router)#router ipx
FastIron-C(config-ospf-router)#vlan 2 name IP-Subnet_1.1.9.0/24
FastIron-C(config-vlan-2)#untag e 1 to 4
FastIron-C(config-vlan-2)#no spanning-tree
FastIron-C(config-vlan-2)#router-interface ve1
FastIron-C(config-vlan-2)#other-proto name block-other-protocols
```

```
FastIron-C(config-vlan-other-proto)#no dynamic
FastIron-C(config-vlan-other-proto)#exclude e 1 to 4
FastIron-C(config-vlan-other-proto)#int ve1
FastIron-C(config-vif-1)#ip addr 1.1.9.1/24
FastIron-C(config-vif-1)#ip ospf area 0.0.0.0
FastIron-C(config-vif-1)#vlan 8 name IPX_Network9
FastIron-C(config-vlan-8)#untag e 5 to 8
FastIron-C(config-vlan-8)#no span
FastIron-C(config-vlan-8)#router-int ve2
FastIron-C(config-vlan-8)#other-proto name block-other-protocols
FastIron-C(config-vlan-other-proto)#no dynamic
FastIron-C(config-vlan-other-proto)#exclude e 5 to 8
FastIron-C(config-vlan-other-proto)#int ve2
FastIron-C(config-vif-2)#ipx net 9 ethernet_802.3
FastIron-C(config-vif-2)#vlan 3 name IP_Sub_&_IPX_Net_VLAN
FastIron-C(config-vlan-3)#untag e 9 to 16
FastIron-C(config-vlan-3)#no spanning-tree
FastIron-C(config-vlan-3)#ip-subnet 1.1.10.0/24
FastIron-C(config-vlan-ip-subnet)#static e 9 to 12
FastIron-C(config-vlan-ip-subnet)#router-interface ve3
FastIron-C(config-vlan-ip-subnet)#ipx-network 10 ethernet_802.3
FastIron-C(config-vlan-ipx-network)#static e 13 to 16
FastIron-C(config-vlan-ipx-network)#router-interface ve4
FastIron-C(config-vlan-ipx-network)#other-proto name block-other-protocols
FastIron-C(config-vlan-other-proto)#exclude e 9 to 16
FastIron-C(config-vlan-other-proto)#no dynamic
FastIron-C(config-vlan-other-proto)#interface ve 3
FastIron-C(config-vif-3)#ip addr 1.1.10.1/24
FastIron-C(config-vif-3)#ip ospf area 0.0.0.0
FastIron-C(config-vif-3)#int ve4
FastIron-C(config-vif-4)#ipx network 10 ethernet_802.3
FastIron-C(config-vif-4)#vlan 4 name Bridged_ALL_Protocols
FastIron-C(config-vlan-4)#untag ethernet 17 to 24
FastIron-C(config-vlan-4)#tag ethernet 25 to 26
FastIron-C(config-vlan-4)#spanning-tree
FastIron-C(config-vlan-4)#vlan 7 name Rtr_BB_to_Bldg.2
FastIron-C(config-vlan-7)#tag e 25
FastIron-C(config-vlan-7)#no spanning-tree
FastIron-C(config-vlan-7)#router-interface ve5
FastIron-C(config-vlan-7)#vlan 6 name Rtr_BB_to_Bldg.1
FastIron-C(config-vlan-6)#tag ethernet 26
FastIron-C(config-vlan-6)#no spanning-tree
FastIron-C(config-vlan-6)#router-interface ve6
FastIron-C(config-vlan-6)#int ve5
FastIron-C(config-vif-5)#ip addr 1.1.8.2/24
FastIron-C(config-vif-5)#ip ospf area 0.0.0.0
FastIron-C(config-vif-5)#ipx network 8 ethernet_802.3
FastIron-C(config-vif-5)#int ve6
FastIron-C(config-vif-6)#ip addr 1.1.5.2/24
FastIron-C(config-vif-6)#ip ospf area 0.0.0.0
```

```
FastIron-C(config-vif-6)#ipx network 5 ethernet_802.3
FastIron-C(config-vif-6)#
```

Configuring Protocol VLANs With Dynamic Ports

The configuration examples for protocol VLANs in the sections above show how to configure the VLANs using static ports. You also can configure the following types of protocol VLANs with dynamic ports:

- AppleTalk protocol
- IP protocol
- IPX protocol
- IP subnet
- IPX network

NOTE: The software does not support dynamically adding ports to AppleTalk cable VLANs. Conceptually, an AppleTalk cable VLAN consists of a single network cable, connected to a single port. Therefore, dynamic addition and removal of ports is not applicable.

NOTE: You cannot route to or from protocol VLANs with dynamically added ports.

Aging of Dynamic Ports

Platform Support:

- FESX/FSX/FWSX devices running software release 02.5.00 and later

When you add the ports to the VLAN, the software automatically adds them all to the VLAN. However, dynamically added ports age out. If the age time for a dynamic port expires, the software removes the port from the VLAN. If that port receives traffic for the IP subnet or IPX network, the software adds the port to the VLAN again and starts the aging timer over. Each time the port receives traffic for the VLAN's IP subnet or IPX network, the aging timer starts over.

NOTE: You can disable VLAN membership aging of dynamically added ports. See “Disabling Membership Aging of Dynamic VLAN Ports” on page 14-34**)**.

Dynamic ports within any protocol VLAN age out after 10 minutes, if no member protocol traffic is received on a port within the VLAN. The aged out port, however, remains as a candidate dynamic port for that VLAN. The port becomes active in the VLAN again if member protocol traffic is received on that port.

Once a port is re-activated, the aging out period for the port is reset to 20 minutes. Each time a member protocol packet is received by a candidate dynamic port (aged out port) the port becomes active again and the aging out period is reset for 20 minutes.

Disabling Membership Aging of Dynamic VLAN Ports

Platform Support:

- FESX/FSX/FWSX devices running software release 02.5.00 and later

You can disable VLAN membership aging of ports that are dynamically assigned to protocol or subnet-based VLANs. This feature resolves the connectivity issue that may occur in certain configurations when protocol or subnet VLANs are configured with dynamic port membership.

NOTE: This issue does not occur with statically assigned VLAN memberships. Thus, enable this feature only if your configuration includes dynamically assigned VLAN memberships for protocol or subnet VLANs.

To enable this feature, enter commands such as the following:

```
FastIron(config)#vlan 10 by port
FastIron(config-vlan-10)#int e 1/1 to 1/5
FastIron(config-vlan-10)#ip-proto name IP_Prot_VLAN
FastIron(config-vlan-ip-proto)#no-dynamic-aging
FastIron(config-vlan-ip-proto)#write memory
```

These commands create an IP protocol VLAN and disable the VLAN membership aging of ports that are dynamically assigned to the protocol VLAN.

Syntax: [no] no-dynamic-aging

Enter the **no** form of the command to disable this feature after it has been enabled.

By default, VLAN membership of dynamically assigned ports will age out after a period of time if no packets belonging to that protocol or subnet VLAN are received by the CPU.

The output of the **show running-config** command indicates if the no-dynamic-aging feature is enabled for a specific protocol or subnet VLAN.

Configuration Guidelines

- You cannot dynamically add a port to a protocol VLAN if the port has any routing configuration parameters. For example, the port cannot have a virtual routing interface, IP subnet address, IPX network address, or AppleTalk network address configured on it.
- Once you dynamically add a port to a protocol VLAN, you cannot configure routing parameters on the port.
- Dynamic VLAN ports are not required or supported on AppleTalk cable VLANs.

Configuring an IP, IPX, or AppleTalk Protocol VLAN with Dynamic Ports

To configure an IP, IPX, or AppleTalk protocol VLAN with dynamic ports, use the following method.

To configure port-based VLAN 10, then configure an IP protocol VLAN within the port-based VLAN with dynamic ports, enter the following commands such as the following:

```
FastIron(config)#vlan 10 by port
FastIron(config-vlan-10)#untag ethernet 1/1 to 1/6
added untagged port ethe 1/1 to 1/6 to port-vlan 30.
FastIron(config-vlan-10)#ip-proto name IP_Prot_VLAN
FastIron(config-vlan-10)#dynamic
FastIron(config)#write memory
```

Syntax: vlan <vlan-id> by port [name <string>]

Syntax: untagged ethernet [<slotnum>/]<portnum> to [<slotnum>/]<portnum>

or

Syntax: untagged ethernet [<slotnum>/]<portnum> ethernet [<slotnum>/]<portnum>

NOTE: Use the first **untagged** command for adding a range of ports. Use the second command for adding separate ports (not in a range).

Syntax: ip-proto [name <string>]

Syntax: ipx-proto [name <string>]

Syntax: appletalk-cable-vlan <num> [name <string>]

Syntax: dynamic

The procedure is similar for IPX and AppleTalk protocol VLANs. Enter **ipx-proto** or **atalk-proto** instead of **ip-proto**.

Configuring an IP Subnet VLAN with Dynamic Ports

To configure port-based VLAN 10, then configure an IP subnet VLAN within the port-based VLAN with dynamic ports, enter commands such as the following:

```
FastIron(config)#vlan 10 by port name IP_VLAN
FastIron(config-vlan-10)#untag ethernet 1/1 to 1/6
added untagged port ethe 1/1 to 1/6 to port-vlan 10.
FastIron(config-vlan-10)#ip-subnet 1.1.1.0/24 name Mktg-LAN
FastIron(config-vlan-10)#dynamic
FastIron(config)#write memory
```

These commands create a port-based VLAN on chassis ports 1/1 – 1/6 named “Mktg-LAN”, configure an IP subnet VLAN within the port-based VLAN, and then add ports from the port-based VLAN dynamically.

Syntax: vlan <vlan-id> by port [name <string>]

Syntax: untagged ethernet [<slotnum>/]<portnum> to [<slotnum>/]<portnum>

or

Syntax: untagged ethernet [<slotnum>/]<portnum> ethernet [<slotnum>/]<portnum>

NOTE: Use the first **untagged** command for adding a range of ports. Use the second command for adding separate ports (not in a range).

Syntax: ip-subnet <ip-addr> <ip-mask> [name <string>]

or

Syntax: ip-subnet <ip-addr>/<mask-bits> [name <string>]

Syntax: dynamic

Configuring an IPX Network VLAN with Dynamic Ports

To configure port-based VLAN 20, then configure an IPX network VLAN within the port-based VLAN with dynamic ports, enter commands such as the following:

```
FastIron(config)#vlan 20 by port name IPX_VLAN
FastIron(config-vlan-10)#untag ethernet 2/1 to 2/6
added untagged port ethe 2/1 to 2/6 to port-vlan 20.
FastIron(config-vlan-10)#ipx-network abcd ethernet_ii name Eng-LAN
FastIron(config-vlan-10)#dynamic
FastIron(config)#write memory
```

These commands create a port-based VLAN on chassis ports 2/1 – 2/6 named “Eng-LAN”, configure an IPX network VLAN within the port-based VLAN, and then add ports from the port-based VLAN dynamically.

Syntax: vlan <vlan-id> by port [name <string>]

Syntax: untagged ethernet [<slotnum>/]<portnum> to [<slotnum>/]<portnum>

or

Syntax: untagged ethernet [<slotnum>/]<portnum> ethernet [<slotnum>/]<portnum>

NOTE: Use the first **untagged** command for adding a range of ports. Use the second command for adding separate ports (not in a range).

Syntax: ipx-network <network-addr> ethernet_ii | ethernet_802.2 | ethernet_802.3 | ethernet_snap [name <string>]

Syntax: dynamic

Configuring Uplink Ports Within a Port-Based VLAN

Platform Support:

- FESX/FSX/FWSX devices running software release 02.5.00 and later

You can configure a subset of the ports in a port-based VLAN as uplink ports. When you configure uplink ports in a port-based VLAN, the device sends all broadcast and unknown-unicast traffic from a port in the VLAN to the uplink ports, but not to other ports within the VLAN. Thus, the uplink ports provide tighter broadcast control within the VLAN.

This uplink port feature behaves the same as the Private VLAN feature, but with the ability to support tagged ports. This feature also supports two Private VLAN modes: the Primary ports (uplink ports) and Isolated ports (host ports).

For example, if two ports within a port-based VLAN are Gigabit ports attached to the network and the other ports in the VLAN are 10/100 ports attached to clients, you can configure the two ports attached to the network as uplink ports. In this configuration, broadcast and unknown-unicast traffic in the VLAN does not go to all ports in the VLAN. The traffic goes only to the uplink ports. The clients on the network do not receive broadcast and unknown-unicast traffic from other ports, including other clients.

Configuration Considerations

- When this feature is enabled, flooded traffic (unknown unicast, unregistered multicast, and broadcast traffic) is software forwarded.
- This feature should not be enabled with protocol VLANs or private VLANs in the same VLAN.

Configuration Syntax

To configure a port-based VLAN containing uplink ports, enter commands such as the following:

```
FastIron(config)#vlan 10 by port
FastIron(config-vlan-10)#untag ethernet 1/1 to 1/24
FastIron(config-vlan-10)#untag ethernet 2/1 to 2/2
FastIron(config-vlan-10)#uplink-switch ethernet 2/1 to 2/2
```

Syntax: [no] uplink-switch ethernet [<slotnum>/]<portnum> [to [<slotnum>/]<portnum> | ethernet [<slotnum>/]<portnum>]

In this example, 24 ports on a 10/100 module and two Gigabit ports on a Gigabit module are added to port-based VLAN 10. The two Gigabit ports are then configured as uplink ports.

Configuring the Same IP Subnet Address on Multiple Port-Based VLANs

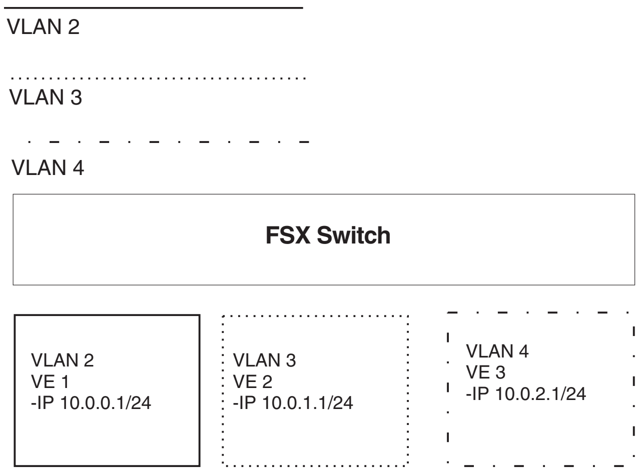
For a Foundry device to route between port-based VLANs, you must add a virtual routing interface to each VLAN. Generally, you also configure a unique IP subnet address on each virtual routing interface. For example, if you have three port-based VLANs, you add a virtual routing interface to each VLAN, then add a separate IP subnet address to each virtual routing interface. The IP address on each of the virtual routing interfaces must be in a separate subnet. The Foundry device routes Layer 3 traffic between the subnets using the subnet addresses.

NOTE: This feature applies only to Layer 3 Switches.

NOTE: Before using the method described in this section, see “Configuring VLAN Groups and Virtual Routing Interface Groups” on page 14-40. You might be able to achieve the results you want using the methods in that section instead.

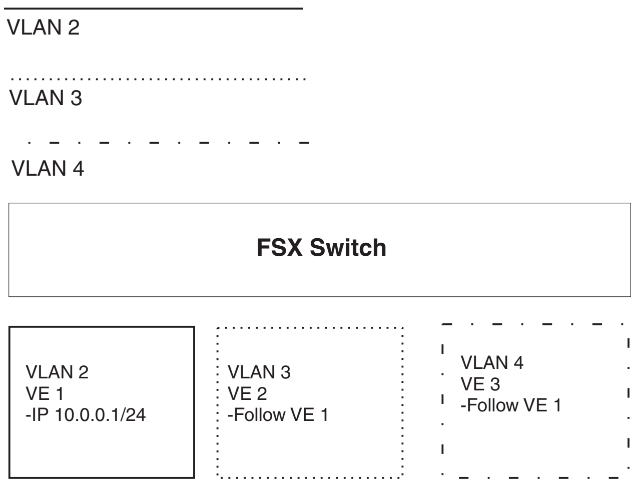
Figure 14.14 shows an example of this type of configuration.

Figure 14.14 Multiple Port-Based VLANs with Separate Protocol Addresses



As shown in this example, each VLAN has a separate IP subnet address. If you need to conserve IP subnet addresses, you can configure multiple VLANs with the same IP subnet address, as shown in Figure 14.15.

Figure 14.15 Multiple Port-Based VLANs with the Same Protocol Address



Each VLAN still requires a separate virtual routing interface. However, all three VLANs now use the same IP subnet address.

In addition to conserving IP subnet addresses, this feature allows containment of Layer 2 broadcasts to segments within an IP subnet. For ISP environments where the same IP subnet is allocated to different customers, placing each customer in a separate VLAN allows all customers to share the IP subnet address, while at the same time isolating them from one another's Layer 2 broadcasts.

NOTE: You can provide redundancy to an IP subnet address that contains multiple VLANs using a pair of Foundry Layer 3 Switches configured for Foundry's VRRP (Virtual Router Redundancy Protocol).

The Foundry device performs proxy Address Resolution Protocol (ARP) for hosts that want to send IP traffic to hosts in other VLANs that are sharing the same IP subnet address. If the source and destination hosts are in the same VLAN, the Foundry device does not need to use ARP.

- If a host attached to one VLAN sends an ARP message for the MAC address of a host in one of the other VLANs using the same IP subnet address, the Foundry device performs a proxy ARP on behalf of the other host. The Foundry device then replies to the ARP by sending the virtual routing interface MAC address. The Foundry device uses the same MAC address for all virtual routing interfaces.

  When the host that sent the ARP then sends a unicast packet addressed to the virtual routing interface's MAC address, the device switches the packet on Layer 3 to the destination host on the VLAN.

  NOTE: If the Foundry device's ARP table does not contain the requested host, the Foundry device forwards the ARP request on Layer 2 to the same VLAN as the one that received the ARP request. Then the device sends an ARP for the destination to the other VLANs that are using the same IP subnet address.

- If the destination is in the same VLAN as the source, the Foundry device does not need to perform a proxy ARP.

To configure multiple VLANs to use the same IP subnet address:

- Configure each VLAN, including adding tagged or untagged ports.
- Configure a separate virtual routing interface for each VLAN, but do not add an IP subnet address to more than one of the virtual routing interfaces.
- Configure the virtual routing interfaces that do not have the IP subnet address to "follow" the virtual routing interface that does have the address.

To configure the VLANs shown in Figure 14.15, you could enter the following commands.

```
FastIron(config)#vlan 1 by port
FastIron(config-vlan-1)#untag ethernet 1/1
FastIron(config-vlan-1)#tag ethernet 1/8
FastIron(config-vlan-1)#router-interface ve 1
```

Syntax: router-interface ve <number>

The commands above configure port-based VLAN 1. The VLAN has one untagged port (1/1) and a tagged port (1/8). In this example, all three VLANs contain port 1/8 so the port must be tagged to allow the port to be in multiple VLANs. You can configure VLANs to share a Layer 3 protocol interface regardless of tagging. A combination of tagged and untagged ports is shown in this example to demonstrate that sharing the interface does not change other VLAN features.

Notice that each VLAN still requires a unique virtual routing interface.

The following commands configure port-based VLANs 2 and 3.

```
FastIron(config-vlan-1)#vlan 2 by port
FastIron(config-vlan-2)#untag ethernet 1/2
FastIron(config-vlan-2)#tag ethernet 1/8
FastIron(config-vlan-2)#router-interface ve 2
FastIron(config-vlan-2)#vlan 3 by port
FastIron(config-vlan-3)#untag ethernet 1/5 to 1/6
FastIron(config-vlan-3)#tag ethernet 1/8
FastIron(config-vlan-3)#router-interface ve 3
```

The following commands configure an IP subnet address on virtual routing interface 1.

```
FastIron(config-vlan-3)#interface ve 1
FastIron(config-vif-1)#ip address 10.0.0.1/24
```

The following commands configure virtual routing interfaces 2 and 3 to "follow" the IP subnet address configured on virtual routing interface 1.

```
FastIron(config-vif-1)#interface ve 2
FastIron(config-vif-2)#ip follow ve 1
FastIron(config-vif-2)#interface ve 3
FastIron(config-vif-3)#ip follow ve 1
```

NOTE: Since virtual routing interfaces 2 and 3 do not have their own IP subnet addresses but instead are "following" virtual routing interface a's IP address, you still can configure an IPX or AppleTalk interface on virtual routing interfaces 2 and 3.

Configuring VLAN Groups and Virtual Routing Interface Groups

To simplify configuration when you have many VLANs with the same configuration, you can configure VLAN groups and virtual routing interface groups.

NOTE: VLAN groups are supported on Layer 3 Switches and Layer 2 Switches. Virtual routing interface groups are supported only on Layer 3 Switches.

When you create a VLAN group, the VLAN parameters you configure for the group apply to all the VLANs within the group. Additionally, you can easily associate the same IP subnet interface with all the VLANs in a group by configuring a virtual routing interface group with the same ID as the VLAN group.

- The VLAN group feature allows you to create multiple port-based VLANs with identical port members. Since the member ports are shared by all the VLANs within the group, you must add the ports as tagged ports. This feature not only simplifies VLAN configuration but also allows you to have a large number of identically configured VLANs in a startup-config file on the device's flash memory module. Normally, a startup-config file with a large number of VLANs might not fit on the flash memory module. By grouping the identically configured VLANs, you can conserve space in the startup-config file so that it fits on the flash memory module.
- The virtual routing interface group feature is useful when you want to configure the same IP subnet address on all the port-based VLANs within a VLAN group. You can configure a virtual routing interface group only after you configure a VLAN group with the same ID. The virtual routing interface group automatically applies to the VLANs in the VLAN group that has the same ID and cannot be applied to other VLAN groups or to individual VLANs.

You can create up to 32 VLAN groups and 32 virtual routing interface groups. A virtual routing interface group always applies only to the VLANs in the VLAN group with the same ID.

NOTE: Depending on the size of the VLAN ID range you want to use for the VLAN group, you might need to allocate additional memory for VLANs. On Layer 3 Switches, if you allocate additional memory for VLANs, you also need to allocate the same amount of memory for virtual routing interfaces. This is true regardless of whether you use the virtual routing interface groups. To allocate additional memory, see "Allocating Memory for More VLANs or Virtual Routing Interfaces" on page 14-42.

Configuring a VLAN Group

To configure a VLAN group, enter commands such as the following:

```
FastIron(config)#vlan-group 1 vlan 2 to 1000
FastIron(config-vlan-group-1)#tagged 1/1 to 1/2
```

The first command in this example begins configuration for VLAN group 1, and assigns VLANs 2 through 1000 to the group. The second command adds ports 1/1 and 1/2 as tagged ports. Since all the VLANs in the group share the ports, you must add the ports as tagged ports.

Syntax: vlan-group <num> vlan <vlan-id> to <vlan-id>

Syntax: tagged ethernet [<slotnum>/]<portnum> [to [<slotnum>/]<portnum> | ethernet [<slotnum>/]<portnum>]

The <num> parameter with the **vlan-group** command specifies the VLAN group ID and can be from 1 – 32. The **vlan** <vlan-id> **to** <vlan-id> parameters specify a contiguous range (a range with no gaps) of individual VLAN IDs. Specify the low VLAN ID first and the high VLAN ID second. The command adds all the specified VLANs to the VLAN group.

NOTE: The device's memory must be configured to contain at least the number of VLANs you specify for the higher end of the range. For example, if you specify 2048 as the VLAN ID at the high end of the range, you first must increase the memory allocation for VLANs to 2048 or higher. Additionally, on Layer 3 Switches, if you allocate additional memory for VLANs, you also need to allocate the same amount of memory for virtual routing interfaces, before you configure the VLAN groups. This is true regardless of whether you use the virtual routing interface groups. The memory allocation is required because the VLAN groups and virtual routing interface groups have a one-to-one mapping. See "Allocating Memory for More VLANs or Virtual Routing Interfaces" on page 14-42.

If a VLAN within the range you specify is already configured, the CLI does not add the group but instead displays an error message. In this case, create the group by specifying a valid contiguous range. Then add more VLANs to the group after the CLI changes to the configuration level for the group. See the following example.

You can add and remove individual VLANs or VLAN ranges from at the VLAN group configuration level. For example, if you want to add VLANs 1001 and 1002 to VLAN group 1 and remove VLANs 900 through 1000, enter the following commands:

```
FastIron(config-vlan-group-1)#add-vlan 1001 to 1002
FastIron(config-vlan-group-1)#remove-vlan 900 to 1000
```

Syntax: add-vlan <vlan-id> [to <vlan-id>]

Syntax: remove-vlan <vlan-id> [to <vlan-id>]

Displaying Information about VLAN Groups

To display VLAN group configuration information, use the **show vlan-group** command.

Syntax: show vlan-group [<group-id>]

The <group-id> specifies a VLAN group. If you do not use this parameter, the configuration information for all the configured VLAN groups is displayed.

Configuring a Virtual Routing Interface Group

A virtual routing interface group allows you to associate the same IP subnet interface with multiple port-based VLANs. For example, if you associate a virtual routing interface group with a VLAN group, all the VLANs in the group have the IP interface of the virtual routing interface group.

Configuration Notes and Feature Limitations

- When you configure a virtual routing interface group, all members of the group have the same IP subnet address. This feature is useful in collocation environments where the device has many IP addresses and you want to conserve the IP address space.
- The **group-router-interface** command creates router interfaces for each VLAN in the VLAN group by using the VLAN IDs of each of the VLANs as the corresponding virtual interface number. Therefore, if a VLAN group contains VLAN IDs greater than the maximum virtual interface number allowed, the **group-router-interface** command will be rejected.

CLI Syntax

To configure a virtual routing interface group, enter commands such as the following:

```
FastIron(config)#vlan-group 1
FastIron(config-vlan-group-1)#group-router-interface
FastIron(config-vlan-group-1)#exit
FastIron(config)#interface group-ve 1
FastIron(config-vif-group-1)#ip address 10.10.10.1/24
```

These commands enable VLAN group 1 to have a group virtual routing interface, then configure virtual routing interface group 1. The software always associates a virtual routing interface group only with the VLAN group that has the same ID. In this example, the VLAN group ID is 1, so the corresponding virtual routing interface group also must have ID 1.

Syntax: group-router-interface

Syntax: interface group-ve <num>

Syntax: [no] ip address <ip-addr> <ip-mask> [secondary]

or

Syntax: [no] ip address <ip-addr>/<mask-bits> [secondary]

The **router-interface-group** command enables a VLAN group to use a virtual routing interface group. Enter this command at the configuration level for the VLAN group. This command configures the VLAN group to use the virtual routing interface group that has the same ID as the VLAN group. You can enter this command when you configure the VLAN group for the first time or later, after you have added tagged ports to the VLAN and so on.

The <num> parameter in the **interface group-ve** <num> command specifies the ID of the VLAN group with which you want to associate this virtual routing interface group. The VLAN group must already be configured and enabled to use a virtual routing interface group. The software automatically associates the virtual routing interface group with the VLAN group that has the same ID. You can associate a virtual routing interface group only with the VLAN group that has the same ID.

NOTE: IPv6 is not supported with **group-ve**.

NOTE: FastIron devices support **group-ve** with OSPF and VRRP protocols only.

The syntax and usage for the **ip address** command is the same as when you use the command at the interface level to add an IP interface.

Displaying the VLAN Group and Virtual Routing Interface Group Information

To verify configuration of VLAN groups and virtual routing interface groups, display the running-config file. If you have saved the configuration to the startup-config file, you also can verify the configuration by displaying the startup-config file. The following example shows the running-config information for the VLAN group and virtual routing interface group configured in the previous examples. The information appears in the same way in the startup-config file.

```
FastIron#show running-config
```

lines not related to the VLAN group omitted...

```
vlan-group 1 vlan 2 to 900
 add-vlan 1001 to 1002
 tagged ethe 1/1 to 1/2
 router-interface-group
```

lines not related to the virtual routing interface group omitted...

```
interface group-ve 1
 ip address 10.10.10.1 255.255.255.0
```

NOTE: If you have enabled display of subnet masks in CIDR notation, the IP address information is shown as follows: 10.10.10.1/24.

Allocating Memory for More VLANs or Virtual Routing Interfaces

Layer 3 Switches can support up to 4095 VLANs and 512 virtual routing interfaces.

The number of VLANs and virtual routing interfaces supported on your product depends on the device and, for Chassis devices, the amount of DRAM on the management module. Table 14.2 lists the default and configurable maximum numbers of VLANs and virtual routing interfaces for Layer 3 Switches and Layer 2 Switches. Unless otherwise noted, the values apply to both types of switches.

Table 14.2: VLAN and Virtual Routing Interface Support

VLANs		**Virtual Routing Interfaces**	
Default Maximum	**Configurable Maximum**	**Default Maximum**	**Configurable Maximum**
64	4094	255	512

NOTE: If many of your VLANs will have an identical configuration, you might want to configure VLAN groups and virtual routing interface groups after you increase the system capacity for VLANs and virtual routing interfaces. See “Configuring VLAN Groups and Virtual Routing Interface Groups” on page 14-40.

Increasing the Number of VLANs You Can Configure

NOTE: Although you can specify up to 4095 VLANs, you can configure only 4094 VLANs. VLAN ID 4094 is reserved for use by the Single Spanning Tree feature.

To increase the maximum number of VLANs you can configure, enter commands such as the following at the global CONFIG level of the CLI:

```
FastIron(config)#system-max vlan 2048
FastIron(config)#write memory
FastIron(config)#end
FastIron#reload
```

Syntax: system-max vlan <num>

The <num> parameter indicates the maximum number of VLANs. The range of valid values depends on the device you are configuring. See Table 14.2.

Increasing the Number of Virtual Routing Interfaces You Can Configure

To increase the maximum number of virtual routing interfaces you can configure, enter commands such as the following at the global CONFIG level of the CLI:

```
FastIron(config)#system-max virtual-interface 512
FastIron(config)#write memory
FastIron(config)#end
FastIron#reload
```

Syntax: system-max virtual-interface <num>

The <num> parameter indicates the maximum number of virtual routing interfaces. The range of valid values depends on the device you are configuring. See Table 14.2.

Configuring Super Aggregated VLANs

You can aggregate multiple VLANs within another VLAN. This feature allows you to construct Layer 2 paths and channels. This feature is particularly useful for Virtual Private Network (VPN) applications in which you need to provide a private, dedicated Ethernet connection for an individual client to transparently reach its subnet across multiple networks.

Conceptually, the paths and channels are similar to Asynchronous Transfer Mode (ATM) paths and channels. A path contains multiple channels, each of which is a dedicated circuit between two end points. The two devices at the end points of the channel appear to each other to be directly attached. The network that connects them is transparent to the two devices.

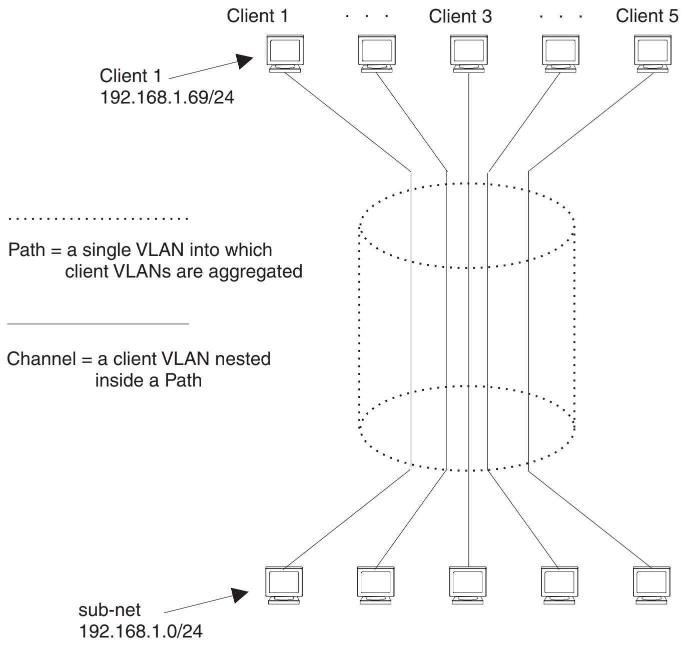
You can aggregate up to 4094 VLANs within another VLAN. This provides a total VLAN capacity on one Foundry device of 16,760,836 channels (4094 * 4094).

The devices connected through the channel are not visible to devices in other channels. Therefore, each client has a private link to the other side of the channel.

The feature allows point-to-point and point-to-multipoint connections.

Figure 14.16 shows a conceptual picture of the service that aggregated VLANs provide. Aggregated VLANs provide a path for multiple client channels. The channels do not receive traffic from other channels. Thus, each channel is a private link.

Figure 14.16 Conceptual Model of the Super Aggregated VLAN Application

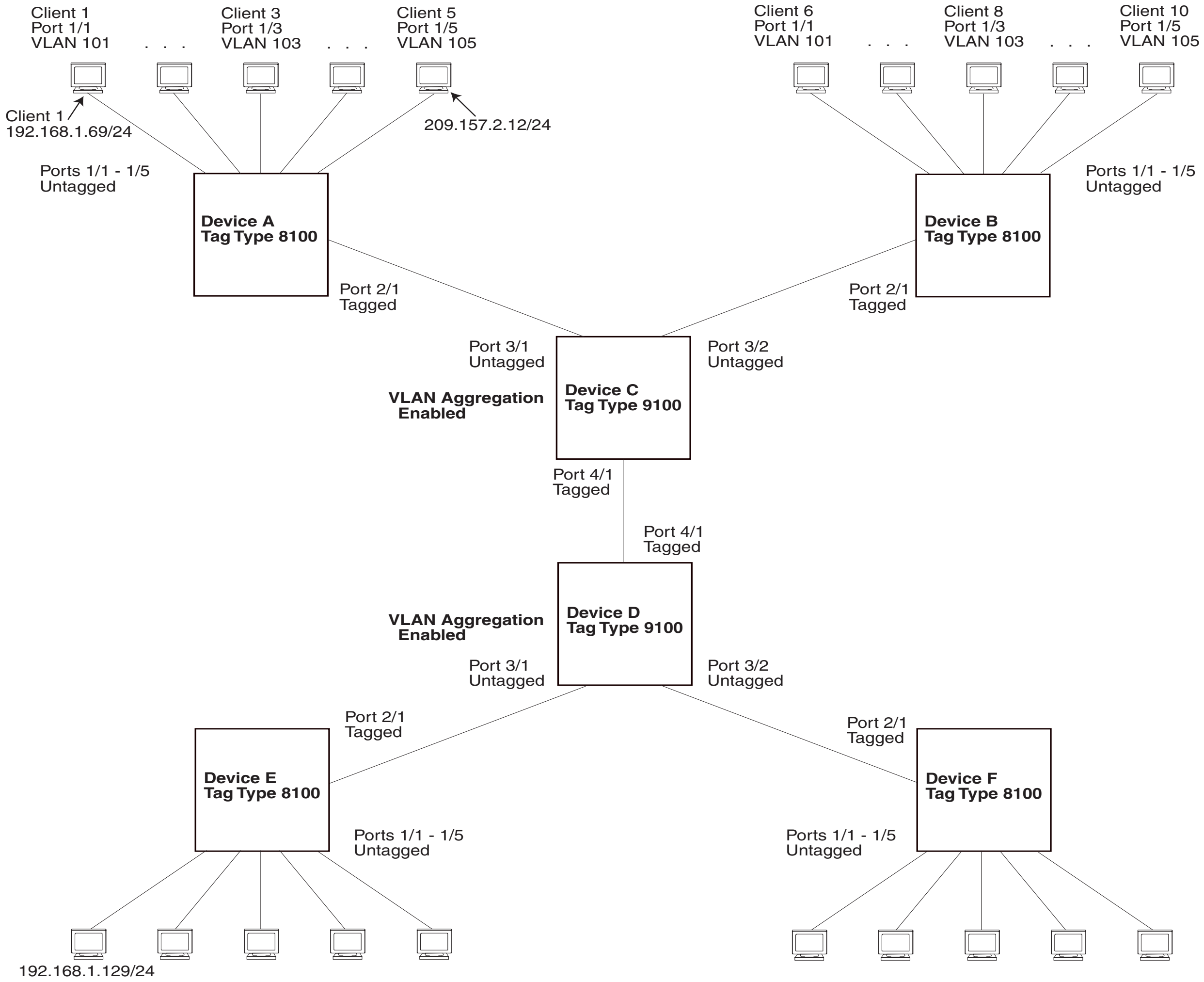

Client 1
. . .
Client 3
. . .
Client 5
Client 1
192.168.1.69/24
Path = a single VLAN into which client VLANs are aggregated
Channel = a client VLAN nested inside a Path
sub-net
192.168.1.0/24

Each client connected to the edge device is in its own port-based VLAN, which is like an ATM channel. All the clients' VLANs are aggregated by the edge device into a single VLAN for connection to the core. The single VLAN that aggregates the clients' VLANs is like an ATM path.

The device that aggregates the VLANs forwards the aggregated VLAN traffic through the core. The core can consist of multiple devices that forward the aggregated VLAN traffic. The edge device at the other end of the core separates the aggregated VLANs into the individual client VLANs before forwarding the traffic. The edge devices forward the individual client traffic to the clients. For the clients' perspective, the channel is a direct point-to-point link.

Figure 14.17 shows an example application that uses aggregated VLANs. This configuration includes the client connections shown in Figure 14.16.

Figure 14.17 Example of a Super Aggregated VLAN Application



In this example, a collocation service provides private channels for multiple clients. Although the same devices are used for all the clients, the VLANs ensure that each client receives its own Layer 2 broadcast domain, separate from the broadcast domains of other clients. For example, client 1 cannot ping client 5.

The clients at each end of a channel appear to each other to be directly connected and thus can be on the same subnet and use network services that require connection to the same subnet. In this example, client 1 is in subnet 192.168.1.0/24 and so is the device at the other end of client 1's channel.

Since each VLAN configured on the core devices is an aggregate of multiple client VLANs, the aggregated VLANs greatly increase the number of clients a core device can accommodate.

This example shows a single link between the core devices. However, you can use a trunk group to add link-level redundancy.

Configuration Rules

- Super Aggregated VLANs and VSRP are not supported together on the same device.

Configuring Aggregated VLANs

To configure aggregated VLANs, perform the following tasks:

- On each edge device, configure a separate port-based VLAN for each client connected to the edge device.

In each client VLAN:

- Add the port connected to the client as an untagged port.
- Add the port connected to the core device (the device that will aggregate the VLANs) as a tagged port. This port must be tagged because all the client VLANs share the port as an uplink to the core device.

- On each core device:
  - Enable VLAN aggregation. This support allows the core device to add an additional tag to each Ethernet frame that contains a VLAN packet from the edge device. The additional tag identifies the aggregate VLAN (the path). However, the additional tag can cause the frame to be longer than the maximum supported frame size. The larger frame support allows Ethernet frames up to 1530 bytes long.

NOTE: Enable the VLAN aggregation option only on the core devices.

- Configure a VLAN tag type (tag ID) that is different than the tag type used on the edge devices. If you use the default tag type (8100) on the edge devices, set the tag type on the core devices to another value, such as 9100. The tag type must be the same on all the core devices. The edge devices also must have the same tag type but the type must be different from the tag type on the core devices.

NOTE: You can enable the Spanning Tree Protocol (STP) on the edge devices or the core devices, but not both. If you enable STP on the edge devices and the core devices, STP will prevent client traffic from travelling through the core to the other side.

Configuring Aggregated VLANs on an Edge Device

To configure the aggregated VLANs on device A in Figure 14.17 on page 14-45, enter the following commands:

```
FastIron(config)#vlan 101 by port
FastIron(config-vlan-101)#tagged ethernet 2/1
FastIron(config-vlan-101)#untagged ethernet 1/1
FastIron(config-vlan-101)#exit
FastIron(config)#vlan 102 by port
FastIron(config-vlan-102)#tagged ethernet 2/1
FastIron(config-vlan-102)#untagged ethernet 1/2
FastIron(config-vlan-102)#exit
FastIron(config)#vlan 103 by port
FastIron(config-vlan-103)#tagged ethernet 2/1
FastIron(config-vlan-103)#untagged ethernet 1/3
FastIron(config-vlan-103)#exit
FastIron(config)#vlan 104 by port
FastIron(config-vlan-104)#tagged ethernet 2/1
FastIron(config-vlan-104)#untagged ethernet 1/4
FastIron(config-vlan-104)#exit
FastIron(config)#vlan 105 by port
FastIron(config-vlan-105)#tagged ethernet 2/1
FastIron(config-vlan-105)#untagged ethernet 1/5
FastIron(config-vlan-105)#exit
FastIron(config)#write memory
```

Syntax: [no] vlan <vlan-id> [by port]

Syntax: [no] tagged ethernet [<slotnum>/]<portnum> [to [<slotnum>/]<portnum> | ethernet [<slotnum>/]<portnum>]

Syntax: [no] untagged ethernet [<slotnum>/]<portnum> [to [<slotnum>/]<portnum> | ethernet [<slotnum>/]<portnum>]

Use the **tagged** command to add the port that the device uses for the uplink to the core device. Use the **untagged** command to add the ports connected to the individual clients.

Configuring Aggregated VLANs on a Core Device

To configure the aggregated VLANs on device C in Figure 14.17 on page 14-45, enter the following commands:

```
FastIron(config)#tag-type 9100
FastIron(config)#aggregated-vlan
FastIron(config)#vlan 101 by port
FastIron(config-vlan-101)#tagged ethernet 4/1
FastIron(config-vlan-101)#untagged ethernet 3/1
FastIron(config-vlan-101)#exit
FastIron(config)#vlan 102 by port
FastIron(config-vlan-102)#tagged ethernet 4/1
FastIron(config-vlan-102)#untagged ethernet 3/2
FastIron(config-vlan-102)#exit
FastIron(config)#write memory
```

Syntax: [no] tag-type <num>

Syntax: [no] aggregated-vlan

The <num> parameter specifies the tag type can be a hexadecimal value from 0 – ffff. The default is 8100.

Verifying the Configuration

You can verify the VLAN, VLAN aggregation option, and tag configuration by viewing the running-config. To display the running-config, enter the **show running-config** command from any CLI prompt. After you save the configuration changes to the startup-config, you also can display the settings in that file by entering the **show configuration** command from any CLI prompt.

Complete CLI Examples

The following sections show all the Aggregated VLAN configuration commands on the devices in Figure 14.17 on page 14-45.

NOTE: In these examples, the configurations of the edge devices (A, B, E, and F) are identical. The configurations of the core devices (C and D) also are identical. The aggregated VLAN configurations of the edge and core devices on one side must be symmetrical (in fact, a mirror image) to the configurations of the devices on the other side. For simplicity, the example in Figure 14.17 on page 14-45 is symmetrical in terms of the port numbers. This allows the configurations for both sides of the link to be the same. If your configuration does not use symmetrically arranged port numbers, the configurations should not be identical but must use the correct port numbers.

Commands for Device A

```
FastIronA(config)#vlan 101 by port
FastIronA(config-vlan-101)#tagged ethernet 2/1
FastIronA(config-vlan-101)#untagged ethernet 1/1
FastIronA(config-vlan-101)#exit
FastIronA(config)#vlan 102 by port
FastIronA(config-vlan-102)#tagged ethernet 2/1
FastIronA(config-vlan-102)#untagged ethernet 1/2
FastIronA(config-vlan-102)#exit
FastIronA(config)#vlan 103 by port
FastIronA(config-vlan-103)#tagged ethernet 2/1
FastIronA(config-vlan-103)#untagged ethernet 1/3
FastIronA(config-vlan-103)#exit
FastIronA(config)#vlan 104 by port
FastIronA(config-vlan-104)#tagged ethernet 2/1
FastIronA(config-vlan-104)#untagged ethernet 1/4
FastIronA(config-vlan-104)#exit
FastIronA(config)#vlan 105 by port
FastIronA(config-vlan-105)#tagged ethernet 2/1
FastIronA(config-vlan-105)#untagged ethernet 1/5
FastIronA(config-vlan-105)#exit
vA(config)#write memory
```

Commands for Device B

The commands for configuring device B are identical to the commands for configuring device A. Notice that you can use the same channel VLAN numbers on each device. The devices that aggregate the VLANs into a path can distinguish between the identically named channel VLANs based on the ID of the path VLAN.

```
FastIronB(config)#vlan 101 by port
FastIronB(config-vlan-101)#tagged ethernet 2/1
FastIronB(config-vlan-101)#untagged ethernet 1/1
FastIronB(config-vlan-101)#exit
FastIronB(config)#vlan 102 by port
FastIronB(config-vlan-102)#tagged ethernet 2/1
FastIronB(config-vlan-102)#untagged ethernet 1/2
FastIronB(config-vlan-102)#exit
FastIronB(config)#vlan 103 by port
FastIronB(config-vlan-103)#tagged ethernet 2/1
FastIronB(config-vlan-103)#untagged ethernet 1/3
FastIronB(config-vlan-103)#exit
FastIronB(config)#vlan 104 by port
FastIronB(config-vlan-104)#tagged ethernet 2/1
FastIronB(config-vlan-104)#untagged ethernet 1/4
FastIronB(config-vlan-104)#exit
FastIronB(config)#vlan 105 by port
FastIronB(config-vlan-105)#tagged ethernet 2/1
FastIronB(config-vlan-105)#untagged ethernet 1/5
FastIronB(config-vlan-105)#exit
FastIronB(config)#write memory
```

Commands for Device C

Since device C is aggregating channel VLANs from devices A and B into a single path, you need to change the tag type and enable VLAN aggregation.

```
FastIronC(config)#tag-type 9100
FastIronC(config)#aggregated-vlan
FastIronC(config)#vlan 101 by port
FastIronC(config-vlan-101)#tagged ethernet 4/1
FastIronC(config-vlan-101)#untagged ethernet 3/1
FastIronC(config-vlan-101)#exit
FastIronC(config)#vlan 102 by port
FastIronC(config-vlan-102)#tagged ethernet 4/1
FastIronC(config-vlan-102)#untagged ethernet 3/2
FastIronC(config-vlan-102)#exit
FastIronC(config)#write memory
```

Commands for Device D

Device D is at the other end of path and separates the channels back into individual VLANs. The tag type must be the same as tag type configured on the other core device (Device C). In addition, VLAN aggregation also must be enabled.

```
FastIronD(config)#tag-type 9100
FastIronD(config)#aggregated-vlan
FastIronD(config)#vlan 101 by port
FastIronD(config-vlan-101)#tagged ethernet 4/1
FastIronD(config-vlan-101)#untagged ethernet 3/1
FastIronD(config-vlan-101)#exit
FastIronD(config)#vlan 102 by port
FastIronD(config-vlan-102)#tagged ethernet 4/1
FastIronD(config-vlan-102)#untagged ethernet 3/2
FastIronD(config-vlan-102)#exit
FastIronD(config)#write memory
```

Commands for Device E

Since the configuration in Figure 14.17 on page 14-45 is symmetrical, the commands for configuring device E are identical to the commands for configuring device A.

```
FastIronE(config)#vlan 101 by port
FastIronE(config-vlan-101)#tagged ethernet 2/1
FastIronE(config-vlan-101)#untagged ethernet 1/1
FastIronE(config-vlan-101)#exit
FastIronE(config)#vlan 102 by port
FastIronE(config-vlan-102)#tagged ethernet 2/1
FastIronE(config-vlan-102)#untagged ethernet 1/2
FastIronE(config-vlan-102)#exit
FastIronE(config)#vlan 103 by port
FastIronE(config-vlan-103)#tagged ethernet 2/1
FastIronE(config-vlan-103)#untagged ethernet 1/3
FastIronE(config-vlan-103)#exit
FastIronE(config)#vlan 104 by port
FastIronE(config-vlan-104)#tagged ethernet 2/1
FastIronE(config-vlan-104)#untagged ethernet 1/4
FastIronE(config-vlan-104)#exit
FastIronE(config)#vlan 105 by port
FastIronE(config-vlan-105)#tagged ethernet 2/1
FastIronE(config-vlan-105)#untagged ethernet 1/5
FastIronE(config-vlan-105)#exit
FastIronE(config)#write memory
```

Commands for Device F

The commands for configuring device F are identical to the commands for configuring device E. In this example, since the port numbers on each side of the configuration in Figure 14.17 on page 14-45 are symmetrical, the configuration of device F is also identical to the configuration of device A and device B.

```
FastIronF(config)#vlan 101 by port
FastIronF(config-vlan-101)#tagged ethernet 2/1
FastIronF(config-vlan-101)#untagged ethernet 1/1
FastIronF(config-vlan-101)#exit
FastIronF(config)#vlan 102 by port
FastIronF(config-vlan-102)#tagged ethernet 2/1
FastIronF(config-vlan-102)#untagged ethernet 1/2
FastIronF(config-vlan-102)#exit
FastIronF(config)#vlan 103 by port
FastIronF(config-vlan-103)#tagged ethernet 2/1
FastIronF(config-vlan-103)#untagged ethernet 1/3
FastIronF(config-vlan-103)#exit
FastIronF(config)#vlan 104 by port
FastIronF(config-vlan-104)#tagged ethernet 2/1
FastIronF(config-vlan-104)#untagged ethernet 1/4
FastIronF(config-vlan-104)#exit
FastIronF(config)#vlan 105 by port
FastIronF(config-vlan-105)#tagged ethernet 2/1
FastIronF(config-vlan-105)#untagged ethernet 1/5
FastIronF(config-vlan-105)#exit
FastIronF(config)#write memory
```

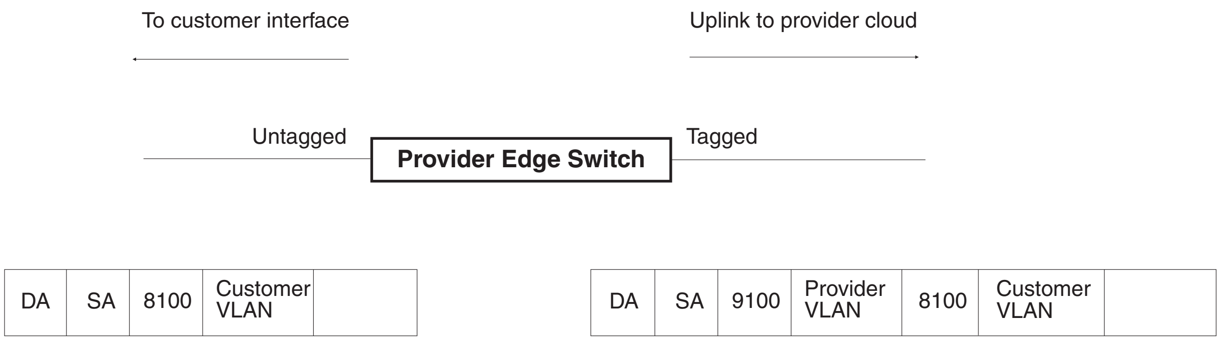
Configuring 802.1Q-in-Q Tagging

802.1Q tagging is an IEEE standard that enables a networking device to add information to a Layer 2 packet in order to identify the VLAN membership of the packet. Foundry devices tag a packet by adding a four-byte tag to the packet. The tag contains the tag value, which identifies the data as a tag, and also contains the VLAN ID of the VLAN from which the packet was sent. The tag and VLAN ID keep traffic from each VLAN segregated and private.

- FESX releases prior to 01.1.00 enable you to configure a single 802.1Q tag type on all ports on the device. The default 802.1Q tag on a Foundry device is 8100 (hexadecimal), compliant with the 802.1Q specification.

  Figure 14.18 shows an 802.1Q configuration example with a single 802.1Q tag type.

Figure 14.18 802.1Q Configuration Example



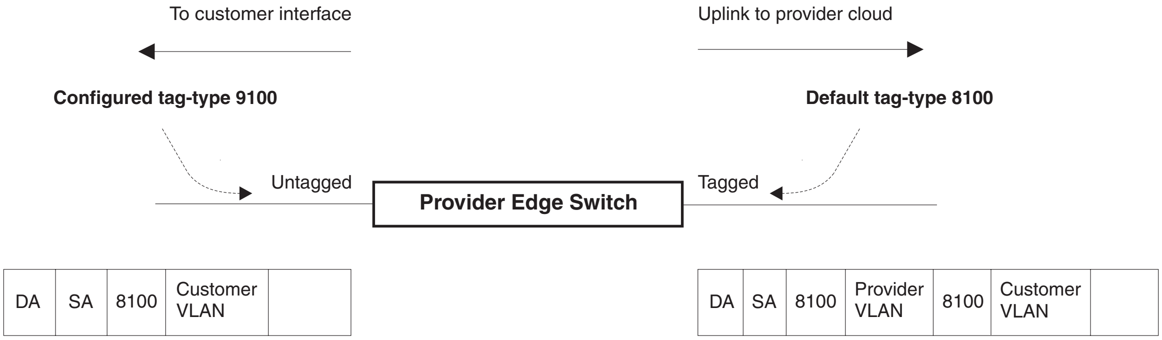
  As shown in Figure 14.18, the ports to customer interfaces are untagged, whereas the uplink ports to the provider cloud are tagged, because multiple client VLANs share the uplink to the provider cloud. In this example, the Foundry device treats the customer's private VLAN ID and 8100 tag type as normal payload, and adds the 9100 tag type to the packet when the packet is sent to the uplink and forwarded along the provider cloud.

  As long as the switches in the provider's network are Foundry devices or devices that can use the 9100 tag type, the data gets switched along the network. However, devices along the provider's cloud that do not support the 9100 tag type may not properly handle the packets.

- FESX releases 01.1.00 and later, and all FSX and FWSX releases, provide finer granularity for configuring 802.1Q tagging, enabling you to configure 802.1Q tag-types on a group of ports. This type of configuration is called ***802.1Q-in-Q tagging***. This feature enables the creation of two identical 802.1Q tags (802.1Q-in-Q tagging) on a single device. This enhancement improves SAV interoperability between Foundry devices and other vendors' devices that support the 802.1Q tag-types, but are not very flexible with the tag-types they accept.

  Figure 14.19 shows an example application with 802.1Q-in-Q tagging.

Figure 14.19 802.1Q-in-Q Configuration Example



In Figure 14.19, the untagged ports (to customer interfaces) accept frames that have any 802.1Q tag other than the configured tag-type 9100. These packets are considered untagged on this incoming port and are re-tagged when they are sent out of the uplink towards the provider. The 802.1Q tag-type on the uplink port is 8100, so the Foundry device will switch the frames to the uplink device with an additional 8100 tag, thereby supporting devices that only support this method of VLAN tagging.

Configuration Rules

- Since the uplink (to the provider cloud) and the edge link (to the customer port) must have different 802.1Q tags, make sure the uplink and edge link are in different port regions. See “About Port Regions” on page 8-1 for a list of valid port regions.
- If you configure a port with an 802.1Q tag-type, the Foundry device automatically applies the 802.1Q tag-type to all ports within the same port region. Likewise, if you remove the 802.1Q tag-type from a port, the Foundry device automatically removes the 802.1Q tag-type from all ports within the same port region.
- X Series devices support one configured tag-type per device along with the default tag-type of 8100. For example, if you configure an 802.1Q tag of 9100 on ports 1 – 12, then later configure an 802.1Q tag of 5100 on port 15, the device automatically applies the 5100 tag to all ports in the same port region as port 15, and also changes the 802.1Q tag-type on ports 1 – 12 to 5100.
- 802.1Q-in-Q tagging and VSRP are not supported together on the same device.

Enabling 802.1Q-in-Q Tagging

To enable 802.1Q-in-Q tagging, configure an 802.1Q tag on the untagged edge links (the customer ports) to any value other than the 802.1Q tag for incoming traffic. For example, in Figure 14.20, the 802.1Q tag on the untagged edge links (ports 11 and 12) is 9100, whereas, the 802.1Q tag for incoming traffic is 8100.

To configure 802.1 Q-in-Q tagging as shown in Figure 14.20, enter commands such as the following on the untagged edge links of devices C and D:

```
FastIron(config)#tag-type 9100 e 11 to 12
FastIron(config)#aggregated-vlan
```

Note that since ports 11 and 12 belong to the port region 1 – 12, the 802.1Q tag actually applies to ports 1 – 12.

Syntax: [no] tag-type <num> [ethernet [<slotnum>/] <port number> [to <port number>]]

The **<num>** parameter specifies the tag-type number and can be a hexadecimal value from 0 - ffff. The default is 8100.

The **<slotnum>** parameter is required on chassis devices.

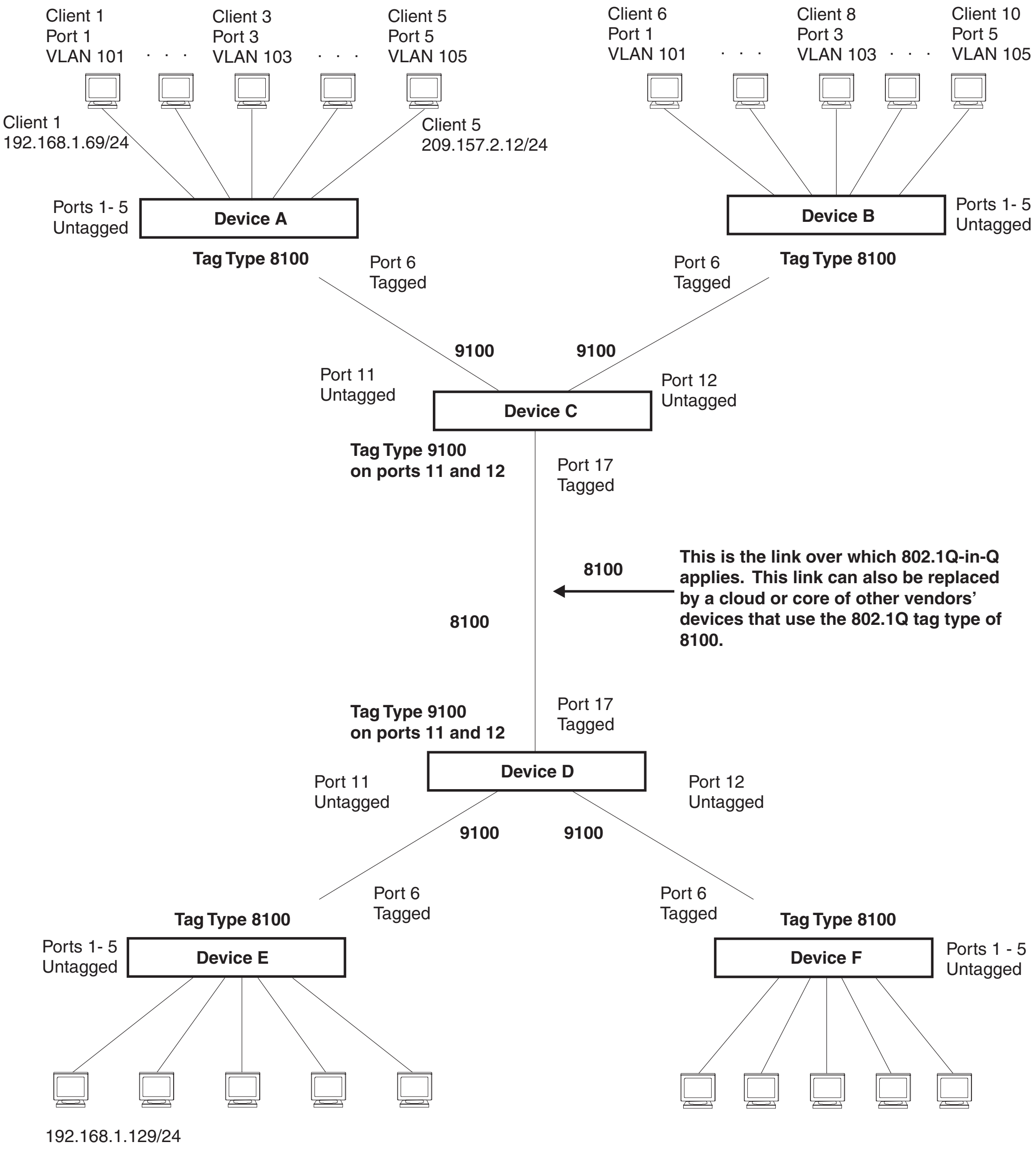
The **ethernet <port number> to <port number>** parameter specifies the port(s) that will use the defined 802.1Q tag. This parameter operates with the following rules:

- If you specify a single port number, the 802.1Q tag applies to all ports within the port region. For example, if you enter the command **tag-type 9100 e 1**, the Foundry device automatically applies the 802.1Q tag to ports 1 – 12 since all of these ports are in the same port region. You can use the **show running-config** command to view how the command has been applied.
- If you do not specify a port or range of ports, the 802.1Q tag applies to all Ethernet ports on the device.

Example Configuration

Figure 14.20 shows an example 802.1Q-in-Q configuration.

Figure 14.20 Example 802.1Q-in-Q Configuration



This is the link over which 802.1Q-in-Q applies. This link can also be replaced by a cloud or core of other vendors' devices that use the 802.1Q tag type of 8100.

Configuring Private VLANs

Platform Support:

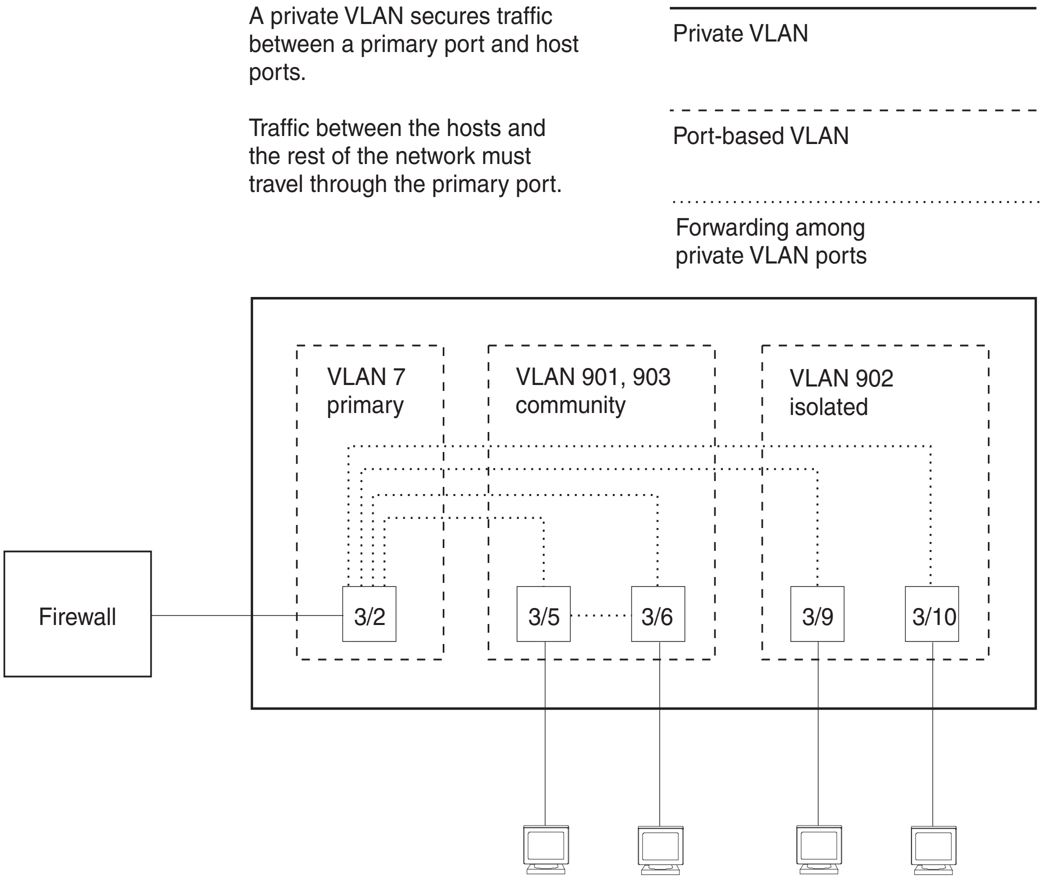
- FastIron X Series devices running software release 02.4.00 and later

---

NOTE: Software releases 02.4.00 and later support private VLANs on untagged ports. You cannot configure isolated, community, or primary VLANs on 802.1Q tagged ports.

A private VLAN is a VLAN that has the properties of standard Layer 2 port-based VLANs but also provides additional control over flooding packets on a VLAN. Figure 14.21 shows an example of an application using a private VLAN.

Figure 14.21 Private VLAN Used to Secure Communication between a Workstation and Servers


A private VLAN secures traffic between a primary port and host ports.
Traffic between the hosts and the rest of the network must travel through the primary port.
Private VLAN
Port-based VLAN
Forwarding among private VLAN ports
VLAN 7 primary
VLAN 901, 903 community
VLAN 902 isolated
Firewall
3/2
3/5
3/6
3/9
3/10

This example uses a private VLAN to secure traffic between hosts and the rest of the network through a firewall. Five ports in this example are members of a private VLAN. The first port (port 3/2) is attached to a firewall. The next four ports (ports 3/5, 3/6, 3/9, and 3/10) are attached to hosts that rely on the firewall to secure traffic between the hosts and the rest of the network. In this example, two of the hosts (on ports 3/5 and 3/6) are in a community private VLAN, and thus can communicate with one another as well as through the firewall. The other two hosts (on ports 3/9 and 3/10), are in an isolated VLAN and thus can communicate only through the firewall. The two hosts are secured from communicating with one another even though they are in the same VLAN.

By default, the private VLAN does not forward broadcast or unknown-unicast packets from outside sources into the private VLAN. If needed, you can override this behavior for broadcast packets, unknown-unicast packets, or both. (See “Enabling Broadcast or Unknown Unicast Traffic to the Private VLAN” on page 14-56.)

You can configure a combination of the following types of private VLANs:

- Primary – The primary private VLAN ports are “promiscuous”. They can communicate with all the isolated private VLAN ports and community private VLAN ports in the isolated and community VLANs that are mapped to the promiscuous port.
- Isolated – Broadcasts and unknown unicasts received on isolated ports are sent only to the primary port. They are not flooded to other ports in the isolated VLAN.
- Community – Broadcasts and unknown unicasts received on community ports are sent to the primary port and also are flooded to the other ports in the community VLAN.

Each private VLAN must have a primary VLAN. The primary VLAN is the interface between the secured ports and the rest of the network. The private VLAN can have any combination of community and isolated VLANs.

Table 14.3 list the differences between private VLANs and standard VLANs.

Table 14.3: Comparison of Private VLANs and Standard Port-Based VLANs

Forwarding Behavior	Private VLANs	Standard VLANs
All ports within a VLAN constitute a common Layer broadcast domain	No	Yes
Broadcasts and unknown unicasts are forwarded to all the VLAN's ports by default	No (isolated VLAN) Yes (community VLAN)	Yes
Known unicasts	Yes	Yes

Configuration Notes

- Private VLANs are supported in releases 02.4.00 and later on untagged ports only. You cannot configure isolated, community, or primary VLANs on 802.1Q tagged ports.
- Normally, in any port-based VLAN, the Foundry device floods unknown unicast, unregistered multicast, and broadcast packets in hardware, although selective packets, such as IGMP, may be sent to only to the CPU for analysis, based on the IGMP snooping configuration. When Protocol or Subnet VLANs are enabled, or if Private VLAN mappings are enabled, the Foundry device will flood unknown unicast, unregistered multicast, and broadcast packets in software.
- There is currently no support for IGMP snooping within private VLANs. In order for clients in private VLANs to receive multicast traffic, IGMP snooping must be disabled so that all multicast packets are treated as unregistered packets and are flooded in software to all the ports.
- FastIron X Series devices forward all known unicast traffic in hardware. This differs from the way the BigIron implements private VLANs, in that the BigIron uses the CPU to forward packets on the primary VLAN's "promiscuous" port. In addition, on the BigIron, support for the hardware forwarding in this feature sometimes results in multiple MAC address entries for the same MAC address in the device's MAC address table. On the X Series devices, multiple MAC entries do not appear in the MAC address table because the X Series transparently manages multiple MAC entries in hardware.
- You can configure private VLANs and dual-mode VLAN ports on the same device. However, the dual-mode VLAN ports cannot be members of Private VLANs.
- A primary VLAN can have multiple ports. All these ports are active, but the ports that will be used depends on the private VLAN mappings. Also, secondary VLANs (isolated and community VLANs) can be mapped to multiple primary VLAN ports. For example:

  ```
  pvlan mapping 901 ethernet 1
  pvlan mapping 901 ethernet 2
  pvlan mapping 901 ethernet 3
  ```

- Trunks are not supported on X Series devices when the ports are part of a private VLAN.

Command Syntax

To configure a private VLAN, configure each of the component VLANs (isolated, community, and public) as a separate port-based VLAN.

- Use standard VLAN configuration commands to create the VLAN and add ports.
- Identify the private VLAN type (isolated, community, or public)

- For the primary VLAN, map the other private VLANs to the port(s) in the primary VLAN

Configuring an Isolated or Community Private VLAN

To configure a community private VLAN, enter commands such as the following:

```
FastIron(config)#vlan 901
FastIron(config-vlan-901)#untagged ethernet 3/5 to 3/6
FastIron(config-vlan-901)#pvlan type community
```

These commands create port-based VLAN 901, add ports 3/5 and 3/6 to the VLAN as untagged ports, then specify that the VLAN is a community private VLAN.

Syntax: untagged ethernet [<slotnum>/]<portnum> [to [<slotnum>/]<portnum> | ethernet [<slotnum>/]<portnum>]

Syntax: [no] pvlan type community | isolated | primary

The **untagged** command adds the ports to the VLAN.

The **pvlan type** command specifies that this port-based VLAN is a private VLAN.

- **community** – Broadcasts and unknown unicasts received on community ports are sent to the primary port and also are flooded to the other ports in the community VLAN.
- **isolated** – Broadcasts and unknown unicasts received on isolated ports are sent only to the primary port. They are not flooded to other ports in the isolated VLAN.
- **primary** – The primary private VLAN ports are “promiscuous”. They can communicate with all the isolated private VLAN ports and community private VLAN ports in the isolated and community VLANs that are mapped to the promiscuous port.

Configuring the Primary VLAN

NOTE: The primary private VLAN has only one active port. If you configure the VLAN to have more than one port, the lowest-numbered port is the active one. The additional ports provide redundancy. If the active port becomes unavailable, the lowest-numbered available port becomes the active port for the VLAN.

To configure a primary private VLAN, enter commands such as the following:

```
FastIron(config)#vlan 7
FastIron(config-vlan-7)#untagged ethernet 3/2
FastIron(config-vlan-7)#pvlan type primary
FastIron(config-vlan-7)#pvlan mapping 901 ethernet 3/2
```

These commands create port-based VLAN 7, add port 3/2 as an untagged port, identify the VLAN as the primary VLAN in a private VLAN, and map the other private VLANs to the port(s) in this VLAN.

Syntax: untagged ethernet [<slotnum>/]<portnum> [to [<slotnum>/]<portnum> | ethernet [<slotnum>/]<portnum>]

Syntax: [no] pvlan type community | isolated | primary

Syntax: [no] pvlan mapping <vlan-id> ethernet [<slotnum>/]<portnum>

The **untagged** command adds the port(s) to the VLAN.

The **pvlan type** command specifies that this port-based VLAN is a private VLAN. Specify **primary** as the type.

The **pvlan mapping** command identifies the other private VLANs for which this VLAN is the primary. The command also specifies the primary VLAN ports to which you are mapping the other private VLANs.

- The <vlan-id> parameter specifies another private VLAN. The other private VLAN you want to specify must already be configured.
- The **ethernet** <portnum> parameter specifies the primary VLAN port to which you are mapping all the ports in the other private VLAN (the one specified by <vlan-id>).

Enabling Broadcast or Unknown Unicast Traffic to the Private VLAN

To enhance private VLAN security, the primary private VLAN does not forward broadcast or unknown unicast packets to its community and isolated VLANs. For example, if port 3/2 in Figure 14.21 on page 14-53 receives a broadcast packet from the firewall, the port does not forward the packet to the other private VLAN ports (3/5, 3/6, 3/9, and 3/10).

This forwarding restriction does not apply to traffic from the private VLAN. The primary port does forward broadcast and unknown unicast packets that are received from the isolated and community VLANs. For example, if the host on port 3/9 sends an unknown unicast packet, port 3/2 forwards the packet to the firewall.

If you want to remove the forwarding restriction, you can enable the primary port to forward broadcast or unknown unicast traffic, if desired, using the following CLI method. You can enable or disable forwarding of broadcast or unknown unicast packets separately.

NOTE: On Layer 2 Switches and Layer 3 Switches, you also can use MAC address filters to control the traffic forwarded into and out of the private VLAN. In addition, if you are using a Layer 2 Switch, you also can use ACLs.

Command Syntax

To configure the ports in the primary VLAN to forward broadcast or unknown unicast and multicast traffic received from sources outside the private VLAN, enter the following commands at the global CONFIG level of the CLI:

```
FastIron(config)#pvlan-preference broadcast flood
FastIron(config)#pvlan-preference unknown-unicast flood
```

These commands enable forwarding of broadcast and unknown-unicast packets to ports within the private VLAN. To again disable forwarding, enter a command such as the following:

```
FastIron(config)#no pvlan-preference broadcast flood
```

This command disables forwarding of broadcast packets within the private VLAN.

Syntax: [no] pvlan-preference broadcast | unknown-unicast flood

CLI Example for Figure 14.21

To configure the private VLANs shown in Figure 14.21 on page 14-53, enter the following commands:

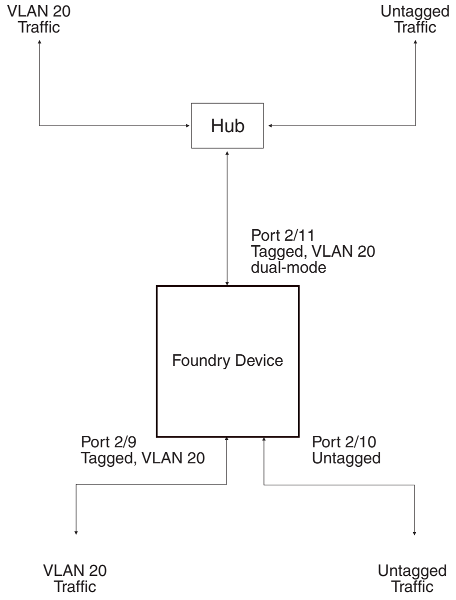
```
FastIron(config)#vlan 901
FastIron(config-vlan-901)#untagged ethernet 3/5 to 3/6
FastIron(config-vlan-901)#pvlan type community
FastIron(config-vlan-901)#exit
FastIron(config)#vlan 902
FastIron(config-vlan-902)#untagged ethernet 3/9 to 3/10
FastIron(config-vlan-902)#pvlan type isolated
FastIron(config-vlan-902)#exit
FastIron(config)#vlan 903
FastIron(config-vlan-903)#untagged ethernet 3/7 to 3/8
FastIron(config-vlan-903)#pvlan type community
FastIron(config-vlan-903)#exit
FastIron(config)#vlan 7
FastIron(config-vlan-7)#untagged ethernet 3/2
FastIron(config-vlan-7)#pvlan type primary
FastIron(config-vlan-7)#pvlan mapping 901 ethernet 3/2
FastIron(config-vlan-7)#pvlan mapping 902 ethernet 3/2
FastIron(config-vlan-7)#pvlan mapping 903 ethernet 3/2
```

Dual-Mode VLAN Ports

Configuring a tagged port as a ***dual-mode*** port allows it to accept and transmit both tagged traffic and untagged traffic at the same time. A dual-mode port accepts and transmits frames belonging to VLANs configured for the port, as well as frames belonging to the default VLAN (that is, untagged traffic).

For example, in Figure 14.22, port 2/11 is a dual-mode port belonging to VLAN 20. Traffic for VLAN 20, as well as traffic for the default VLAN, flows from a hub to this port. The dual-mode feature allows traffic for VLAN 20 and untagged traffic to go through the port at the same time.

Figure 14.22 Dual-Mode VLAN Port Example



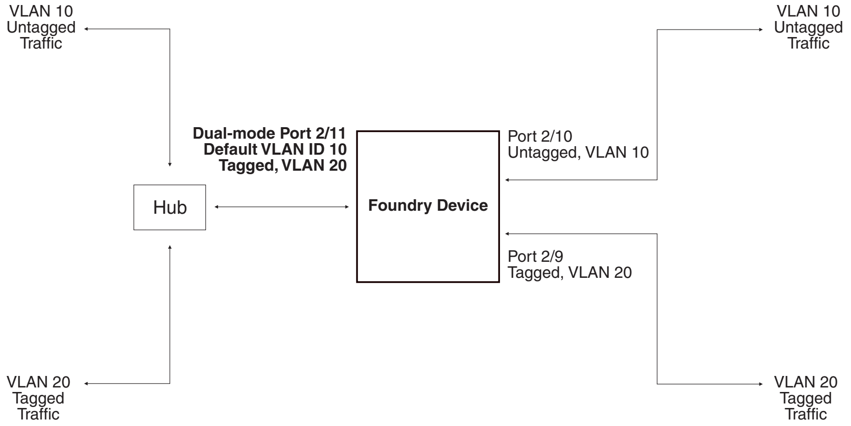
To enable the dual-mode feature on port 2/11 in Figure 14.22:

```
FastIron(config)#vlan 20
FastIron(config-vlan-20)#tagged e 2/11
FastIron(config-vlan-20)#tagged e 2/9
FastIron(config-vlan-20)#int e 2/11
FastIron(config-if-e1000-2/11)#dual-mode
FastIron(config-if-e1000-2/11)#exit
```

Syntax: [no] dual-mode

You can configure a dual-mode port to transmit traffic for a specified VLAN (other than the DEFAULT-VLAN) as untagged, while transmitting traffic for other VLANs as tagged. Figure 14.23 illustrates this enhancement.

Figure 14.23 Specifying a Default VLAN ID for a Dual-Mode Port



In Figure 14.23, tagged port 2/11 is a dual-mode port belonging to VLANs 10 and 20. The default VLAN assigned to this dual-mode port is 10. This means that the port transmits tagged traffic on VLAN 20 (and all other VLANs to which the port belongs) and transmits untagged traffic on VLAN 10.

The dual-mode feature allows tagged traffic for VLAN 20 and untagged traffic for VLAN 10 to go through port 2/11 at the same time. A dual-mode port transmits only untagged traffic on its default VLAN (that is, either VLAN 1, or a user-specified VLAN ID), and only tagged traffic on all other VLANs.

The following commands configure VLANs 10 and 20 in Figure 14.23. Tagged port 2/11 is added to VLANs 10 and 20, then designated a dual-mode port whose specified default VLAN is 10. In this configuration, port 2/11 transmits only untagged traffic on VLAN 10 and only tagged traffic on VLAN 20.

```
FastIron(config)#vlan 10 by port
FastIron(config-vlan-10)#untagged e 2/10
FastIron(config-vlan-10)#tagged e 2/11
FastIron(config-vlan-10)#exit

FastIron(config)#vlan 20 by port
FastIron(config-vlan-20)#tagged e 2/9
FastIron(config-vlan-20)#tagged e 2/11
FastIron(config-vlan-20)#exit

FastIron(config)#int e 2/11
FastIron(config-if-e1000-2/11)#dual-mode 10
FastIron(config-if-e1000-2/11)#exit
```

Syntax: [no] dual-mode [<vlan-id>]

Notes:

- If you do not specify a <vlan-id> in the **dual mode** command, the port's default VLAN is set to 1. The port transmits untagged traffic on the DEFAULT-VLAN.
- The dual-mode feature is disabled by default. Only tagged ports can be configured as dual-mode ports.
- In trunk group, either all of the ports must be dual-mode, or none of them can be.

The **show vlan** command displays a separate row for dual-mode ports on each VLAN. For example:

```
FastIron#show vlan
Total PORT-VLAN entries: 3
Maximum PORT-VLAN entries: 16

legend: [S=Slot]

PORT-VLAN 1, Name DEFAULT-VLAN, Priority level0, Spanning tree Off
 Untagged Ports: (S1)  1  2  3  4  5  6  7  8
 Untagged Ports: (S2)  1  2  3  4  5  6  7  8 12 13 14 15 16 17 18 19
 Untagged Ports: (S2) 20 21 22 23 24
   Tagged Ports: None
   Uplink Ports: None
 DualMode Ports: None
PORT-VLAN 10, Name [None], Priority level0, Spanning tree Off
 Untagged Ports: (S2) 10
   Tagged Ports: None
   Uplink Ports: None
 DualMode Ports: (S2) 11
PORT-VLAN 20, Name [None], Priority level0, Spanning tree Off
 Untagged Ports: None
   Tagged Ports: (S2)  9
   Uplink Ports: None
 DualMode Ports: (S2) 11
```

Displaying VLAN Information

After you configure the VLANs, you can verify the configuration using the following methods.

NOTE: If a VLAN name begins with “GVRP_VLAN_“, the VLAN was created by the GARP VLAN Registration Protocol (GVRP). If a VLAN name begins with “STATIC_VLAN_“, the VLAN was created by GVRP and then was converted into a statically configured VLAN.

Displaying VLANs in Alphanumeric Order

Platform Support:

- FESX/FSX/FWSX devices running software release 03.0.00 and later

In releases prior to 03.0.00 for the FastIron X Series devices, the output of some show commands list VLANs in the order that they were configured. Starting with release 03.0.00, by default, the VLANs are displayed in alphanumeric order.

For example, in releases prior to 03.0.00, if you configure VLANs in the order VLAN 10, VLAN 100, then VLAN 2, the **show run** command output displays:

```
FastIron#show run
...
vlan 10 by port
...
vlan 100 by port
...
vlan 2 by port
...
```

Starting with release 03.0.00, VLANs are displayed in alphanumeric order, as shown in the following example.

```
FastIron#show run
...
vlan 2 by port
...
vlan 10 by port
...
vlan 100 by port
...
```

Displaying System-Wide VLAN Information

Use one of the following methods to display VLAN information for all the VLANs configured on the device.

Enter the following command at any CLI level. This example shows the display for the IP subnet and IPX network VLANs configured in the examples in “Configuring an IP Subnet VLAN with Dynamic Ports” on page 14-36 and “Configuring an IPX Network VLAN with Dynamic Ports” on page 14-36.

```
FastIron#show vlans

Total PORT-VLAN entries: 2
Maximum PORT-VLAN entries: 8
legend: [S=Slot]

PORT-VLAN 1, Name DEFAULT-VLAN, Priority level0, Spanning tree Off
 Untagged Ports: (S2)  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16
 Untagged Ports: (S2) 17 18 19 20 21 22 23 24
 Untagged Ports: (S4)  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16
 Untagged Ports: (S4) 17 18 19 20 21 22 23 24
   Tagged Ports: None

PORT-VLAN 10, Name IP_VLAN, Priority level0, Spanning tree Off
 Untagged Ports: (S1)  1  2  3  4  5  6
   Tagged Ports: None

 IP-subnet VLAN 1.1.1.0 255.255.255.0, Dynamic port enabled
          Name: Mktg-LAN
  Static ports: None
 Exclude ports: None
 Dynamic ports: (S1)  1  2  3  4  5  6
 PORT-VLAN 20, Name IPX_VLAN, Priority level0, Spanning tree Off
 Untagged Ports: (S2)  1  2  3  4  5  6
   Tagged Ports: None

 IPX-network VLAN 0000ABCD, frame type ethernet_ii, Dynamic port enabled
          Name: Eng-LAN
  Static ports: None
 Exclude ports: None
 Dynamic ports: (S2)  1  2  3  4  5  6
```

Syntax: show vlans [<vlan-id> | ethernet [<slotnum>/]<portnum>]

The <vlan-id> parameter specifies a VLAN for which you want to display the configuration information.

The <slotnum> parameter is required on chassis devices.

The <portnum> parameter specifies a port. If you use this parameter, the command lists all the VLAN memberships for the port.

Displaying VLAN Information for Specific Ports

Use one of the following methods to display VLAN information for specific ports.

To display VLAN information for all the VLANs of which port 7/1 is a member, enter the following command:

```
FastIron#show vlans e 7/1

Total PORT-VLAN entries: 3
Maximum PORT-VLAN entries: 8

legend: [S=Slot]

PORT-VLAN 100, Name [None], Priority level0, Spanning tree Off
 Untagged Ports: (S7)  1  2  3  4
   Tagged Ports: None

 IP-subnet VLAN 207.95.11.0 255.255.255.0, Dynamic port disabled
 Static ports: (S7)  1  2
 Exclude ports: None
 Dynamic ports: None
```

Syntax: show vlans [<vlan-id> | ethernet [<slotnum>/]<portnum>

The <vlan-id> parameter specifies a VLAN for which you want to display the configuration information.

The <slotnum> parameter is required on chassis devices.

The <portnum> parameter specifies a port. If you use this parameter, the command lists all the VLAN memberships for the port.

```
FastIron#show vlan-group
vlan-group 1 vlan 2 to 20
 tagged ethe 1/1 to 1/2
!
vlan-group 2 vlan 21 to 40
 tagged ethe 1/1 to 1/2
!
```

Chapter 15

Configuring MAC-Based VLANs

This chapter describes how to configure MAC-Based VLANs on FastIron GS and FastIron LS Layer 2 Switches using the CLI.

Overview

Platform Support:

- FGS and FLS devices running software release 03.0.00 and later

The MAC-based VLAN feature controls network access by authenticating a host's source MAC address, and mapping the incoming packet source MAC to a VLAN. Mapping is based on the MAC address of the end station connected to the physical port. Users who relocate can remain on the same VLAN as long as they connect to any switch in the same domain, on a port which is permitted in the VLAN. The MAC-Based VLAN feature may be enabled for two types of hosts: static and dynamic.

MAC-Based VLAN activity is determined by authentication through a RADIUS server. Incoming traffic that originates from a specific MAC address is forwarded only if the source MAC address-to-VLAN mapping is successfully authenticated. While multi-device port authentication is in progress, all traffic from the new MAC address will be blocked or dropped until the authentication succeeds. Traffic is dropped if the authentication fails.

Static and Dynamic Hosts

Static hosts are devices on the network that do not speak until spoken to. Static hosts may not initiate a request for authentication on their own. Such static hosts can be managed through a **link up** or **link down** notification.

Dynamic hosts are "chatty" devices that generate packets whenever they are in the **link up** state. Dynamic hosts must be authenticated before they can switch or forward traffic.

MAC-Based VLAN Feature Structure

The MAC-Based VLAN feature operates in two stages:

- Source MAC Address Authentication
- Policy-Based Classification and Forwarding

Source MAC Address Authentication

Source MAC address authentication is performed by a central RADIUS server when it receives a PAP request with a username and password that match the MAC address being authenticated. When the MAC address is successfully authenticated, the server must return the VLAN identifier, which is carried in the Tunnel-Type, Tunnel-Medium-Type, and Tunnel-Private-Group-ID attributes of the RADIUS packets. If the Tunnel-Type is *tagged*, the MAC address will be blocked or restricted. If the identified VLAN does not exist, then the authentication is considered a failure, and action is taken based on the configured failure options. (The default failure action is to drop the traffic.) The RADIUS server may also optionally return the QoS attribute for the authenticated MAC address. See Table 15.2 on page 15-4 for more information about attributes.

Policy-Based Classification and Forwarding

Once the authentication stage is complete, incoming traffic is classified based on the response from the RADIUS server. There are three possible actions:

- Incoming traffic from a specific source MAC is dropped because authentication failed
- Incoming traffic from a specific source MAC is classified as untagged into a specific VLAN
- Incoming traffic from a specific source MAC is classified as untagged into a restricted VLAN

Traffic classification is performed by programming incoming traffic and RADIUS-returned attributes in the hardware. Incoming traffic attributes include the source MAC address and the port on which the feature is enabled. The RADIUS-returned attributes are the VLAN into which the traffic is to be classified, and the QoS priority.

NOTE: This feature drops any incoming *tagged* traffic on the port, and classifies and forwards untagged traffic into the appropriate VLANs.

This feature supports up to a maximum of 32 MAC addresses per physical port, with a default of 2. Once a client MAC address is successfully authenticated and registered, the MAC-to-VLAN association remains until the port connection is dropped, or the MAC entry expires.

MAC-Based VLAN can work together with MAC filters or ACL filters on the same port. Regardless of the configuration sequence used, MAC-Based VLAN will always take precedence. For example:

- A user can classify an incoming source MAC into VLAN A and rate-limit the forwarding traffic with an ACL filter, or port-based rate limiting.
- A user can classify an incoming source MAC into VLAN A and deny the traffic based on the destination MAC.

MAC-Based VLAN and Port Up/Down Events

When the state of a port is changed to *down*, all authorized and unauthorized MAC addresses are removed from the MAC-to-VLAN mapping table, any pending authentication requests are cancelled.

Dynamic MAC-Based VLAN

Platform Support:

- FGS and FLS devices running software release 04.0.00 and later

When enabled, this feature allows the dynamic addition of mac-vlan-permit ports to the VLAN table only after successful RADIUS authentication. Ports that fail RADIUS authentication are not added to the VLAN table.

When this feature is not enabled, the physical port is statically added to the hardware table, regardless of the outcome of the authentication process. This feature prevents the addition of un-authenticated ports to the VLAN table. For information about how to configure Dynamic MAC-Based VLAN, see “Configuring Dynamic MAC-Based VLAN” on page 15-7.

Configuring MAC-Based VLANs

Configure MAC-Based VLAN mapping on the switch statically for static hosts, or dynamically for non-static hosts, by directing the RADIUS server to authenticate the incoming packet.

To configure the a MAC-Based VLAN, first perform the following tasks:

- In the VLANs, configure **mac-vlan-permit** for each port that will be participating in the MAC-Based VLAN
- If a port has been MAC-Based VLAN-enabled, but has **not** been added as **mac-vlan-permit** in any of the VLANs, any MAC addresses learned on this port will be blocked in the reserved VLAN. To prevent this, you

must create all of the VLANs and add all ports as **mac-vlan-permit** ***before*** enabling MAC-Based VLAN on any ports.

- Disable any multi-device port authentication on ports you will be using for MAC-to-VLAN mapping

Using MAC-Based VLANs and 802.1X Security on the Same Port

On Foundry FGS and FLS devices, MAC-based VLANs and 802.1X security can be configured on the same port. When both of these features are enabled on the same port, MAC-Based VLAN is performed prior to 802.1X authentication. If MAC-Based VLAN is successful, 802.1X authentication may be performed, based on the configuration of a vendor-specific attribute (VSA) in the profile for the MAC address on the RADIUS server.

When both features are configured on a port, a device connected to the port is authenticated as follows:

1. MAC-Based VLAN is performed on the device to authenticate the device's MAC address.
2. If MAC-Based VLAN is successful, the device then checks to see if the RADIUS server included the Foundry-802_1x-enable VSA (described in Table 15.2) in the Access-Accept message that authenticated the device.
3. If the Foundry-802_1x-enable VSA is not present in the Access-Accept message, or is present and set to 1, then 802.1X authentication is performed for the device.
4. If the Foundry-802_1x-enable VSA is present in the Access-Accept message, and is set to 0, then 802.1X authentication is skipped.

Configuring Generic and Foundry Vendor-Specific Attributes on the RADIUS Server

If the RADIUS authentication process is successful, the RADIUS server sends an Access-Accept message to the Foundry device, authenticating the device. The Access-Accept message includes Vendor-Specific Attributes (VSAs) that specify additional information about the device.

Add Foundry vendor-specific attributes to your RADIUS server's configuration, and configure the attributes in the individual or group profiles of the devices that will be authenticated. Foundry's vendor-ID is 1991, vendor-type 1. Table 15.1 lists generic RADIUS attributes. Table 15.2 lists Foundry Vendor-Specific Attributes.

Table 15.1: Generic RADIUS Attributes

Attribute Name	Attribute ID	Data Type	Optional or Mandatory	Description
Tunnel-Type	64	13 decimal VLAN	Mandatory	RFC 2868.
Tunnel-Medium-Type	65	6 decimal 802	Mandatory	RFC 2868.
Tunnel-Private-Group-ID	81	decimal	Mandatory	RFC 2868. <vlan-id> or U:<vlan -id> – a MAC-Based VLAN ID configured on the Foundry device.

Table 15.2:Foundry vendor-specific attributes for RADIUS

Attribute Name	Attribute ID	Data Type	Optional or Mandatory	Description
Foundry-MAC-Based VLAN-QoS	8	decimal	Optional	The QoS attribute specifies the priority of the incoming traffic based on any value between 0 (lowest priority) and 7 (highest priority). Default is 0.
Foundry-802_1x-enable	6	integer	Optional	Specifies whether 802.1X authentication is performed when MAC-Based VLAN is successful for a device. This attribute can be set to one of the following: **0** Do not perform 802.1X authentication on a device that passes MAC-Based VLAN. Set the attribute to zero (0) for devices that do not support 802.1X authentication. **1** Perform 802.1X authentication when a device passes MAC-Based VLAN. Set the attribute to one (1) for devices that support 802.1X authentication.
Foundry-802_1x-valid	7	integer	Optional	Specifies whether the RADIUS record is valid only for MAC-Based VLAN, or for both MAC-Based VLAN and 802.1X authentication. This attribute can be set to one of the following: **0** The RADIUS record is valid only for MAC-Based VLAN. Set this attribute to zero (0) to prevent a user from using their MAC address as username and password for 802.1X authentication **1** The RADIUS record is valid for both MAC-Based VLAN and 802.1X authentication.

Aging for MAC-Based VLAN

The aging process for MAC-Based VLAN works as described below.

For Permitted Hosts

For permitted hosts, as long as the Foundry device is receiving traffic aging does not occur. The age column in the output of the **show table-mac-vlan** command displays Ena or S <num>. If the Foundry device stops receiving traffic, the entry first ages out from the MAC table (in the hardware) and then the aging cycle for MAC-Based VLAN begins. Aging in the MAC-Based VLAN continues for 2 minutes (the default is 120 seconds) after which the MAC-Based VLAN session is flushed out.

For Blocked Hosts

For blocked hosts, as long as the Foundry device is receiving traffic, aging does not occur. In the output of the **show table-mac-vlan command**, the age column displays H0 to H70, S0, and H0 to H70, etc. Aging of the MAC-Based VLAN MAC occurs in two phases: hardware aging and software aging. The hardware aging period can be configured using the **mac-authentication hw-deny-age** command in config mode. The default is 70 seconds. The software aging time for MAC-Based VLAN MACs can be configured using the **mac-authentication max-age** command. When the Foundry device is no longer receiving traffic from a MAC-Based VLAN MAC address, the hardware aging period begins and lasts for a fixed length of time (default or user-configured). When the hardware aging period ends, the software aging period begins. The software aging period lasts for a configurable amount of time (the default is 120 seconds). After the software aging period ends, the MAC-Based VLAN session is flushed, and the MAC address can be authenticated or denied if the Foundry device again receives traffic from that MAC address.

For MAC-Based Dynamic Activation

If all of the sessions age out on a port, the port is dynamically removed from the VLAN table. When any new session is established, the port is dynamically added back to the VLAN table.

NOTE: If the Foundry device receives a packet from an authenticated MAC address, and the MAC-Based VLAN software aging is still in progress (hardware aging has already occurred), a RADIUS message is NOT sent to the RADIUS server. Instead the MAC address is reentered in the hardware along with the parameters previously returned from the RADIUS server. A RADIUS message is sent only when the MAC-Based VLAN session ages out from the software.

To Change the Length of the Software Aging Period

To change the length of the software aging period for blocked MAC addresses, enter a command such as the following:

```
FGS648P Switch(config)#mac-authentication max-age 180
```

Syntax: [no] mac-authentication max-age <seconds>

You can specify from 1 – 65535 seconds. The default is 120 seconds.

Disabling Aging for MAC-Based VLAN Sessions

MAC addresses that have been authenticated or denied by a RADIUS server are aged out if no traffic is received from the MAC address for a certain period of time.

You can optionally disable aging for MAC-Based VLAN session subject to authentication, either for all MAC addresses or for those learned on a specified interface.

Globally Disabling Aging

On most devices, you can disable aging on all interfaces where MAC-Based VLAN has been enabled, by entering the following command:

```
FGS648P Switch(config)#mac-authentication disable-aging
```

Syntax: mac-authentication disable-aging

Enter the command at the global or interface configuration level.

The **denied-mac-only** parameter prevents denied sessions from being aged out, but ages out permitted sessions.

The **permitted-mac-only** parameter prevents permitted (authenticated and restricted) sessions from being aged out and ages denied sessions.

Disabling the Aging on Interfaces

To disable aging on a specific interface where MAC-Based VLAN has been enabled, enter the command at the interface level. For example:

```
FGS648P Switch(config)#interface e 3/1
```

```
FGS648P Switch(config-if-e1000-3/1)#mac-authentication disable-aging
```

Syntax: [no] mac-authentication disable-aging

Configuring a MAC-Based VLAN for a Static Host

To configure a MAC-Based VLAN for a static host, perform the following steps:

1. Enable multi-device port authentication globally using the following command:

   ```
   FGS648P Switch(config)#mac-authentication enable
   ```

2. Add each port on which you want MAC-Based VLAN enabled as **mac-vlan-permit** for a specific VLAN:

   ```
   FGS648P Switch(config)#vlan 10 by port
   FGS648P Switch(config-vlan-10)#mac-vlan-permit ethernet 0/1/1 to 0/1/6
   added mac-vlan-permit ports ethe 0/1/1 to 0/1/6 to port-vlan 10.
   ```

3. Add the static MAC-Based VLAN configuration on the port:

   ```
   FGS648P Switch(config)#interface e 0/1/1
   FGS648P Switch(config-if-e1000-0/1/1)#mac-authentication mac-vlan 0000.0010.0011
   vlan 10 priority 5
   ```

4. To enable MAC-Based VLAN on the port:

   ```
   FGS648P Switch(config)#interface e 0/1/1
   FGS648P Switch(config-if-e1000-0/1/1)#mac-authentication mac-vlan enable
   ```

5. To disable MAC-Based VLAN on the port:

   ```
   FGS648P Switch(config)#interface e 0/1/1
   FGS648P Switch(interface-0/1/1)#mac-auth mac-vlan disable
   ```

6. To remove and disable the MAC-Based VLAN configuration:

   ```
   FGS648P Switch(config)#interface e 0/1/1
   FGS648P Switch(config-if-e1000-0/1/1)#no mac-auth mac-vlan
   ```

Configuring MAC-Based VLAN for a Dynamic Host

To configure MAC-Based VLAN for a dynamic host, use the following steps:

1. Enable multi-device port authentication globally using the following command:

   ```
   FGS648P Switch(config)#mac-authentication enable
   ```

2. Add each port on which you want MAC-Based VLAN enabled as **mac-vlan-permit** for a specific VLAN:

   ```
   FGS648P Switch(config)#vlan 10 by port
   FGS648P Switch(config-vlan-10)#mac-vlan-permit ethernet 0/1/1 to 0/1/6
   ```

3. To enable MAC-Based VLAN on the port:

   ```
   FGS648P Switch(config)#interface e 0/1/1
   FGS648P Switch(config-if-e1000-0/1/1)#mac-authentication mac-vlan enable
   ```

4. To disable MAC-Based VLAN on the port:

   ```
   FGS648P Switch(config)#interface e 0/1/1
   FGS648P Switch(config-if-e1000-0/1/1)#mac-auth mac-vlan disable
   ```

5. To remove and disable the MAC-Based VLAN configuration:

   ```
   FGS648P Switch(config)#interface e 0/1/1
   FGS648P Switch(config-if-e1000-0/1/1)#no mac-auth mac-vlan
   ```

Configuring the Maximum MAC Addresses per Port

To configure the maximum number of MAC addresses allowed per port, use the following steps:

1. Add the desired maximum MAC entries per port:

```
FGS648P Switch(config)#interface e 0/1/1
FGS648P Switch(config-if-e1000-0/1/1)#mac-authentication mac-vlan max-mac-
entries 24
```

NOTE: 32 MACs maximum are allowed per port. This total includes both static and dynamic hosts. The default number of allowed MACs is 2.

Configuring Dynamic MAC-Based VLAN

To globally enable MAC-Based VLAN globally (for all MAC-based VLAN ports), enter the following commands:

```
FGS648P Switch(config)#mac-authentication enable
FGS648P Switch(config)#mac-authentication mac-vlan-dyn-activation
```

To configure Dynamic MAC-based VLAN to add a specific port to a specific VLAN, enter commands similar to the following:

```
FGS648P Switch(config)#vlan 10
FGS648P Switch(config-vlan-10)#mac-vlan-permit e 0/1/35
```

Syntax: mac-vlan-permit < stack | slot | port>

To disable Dynamic MAC-Based VLAN, enter the following command:

```
FGS648P Switch(config)#no mac-authentication mac-vlan-dyn-activation
```

NOTE: If static Mac-Based VLAN is configured on a port, the port will be added only to the VLAN table for which the static MAC-Based VLAN configuration exists.

NOTE: If the Dynamic MAC-Based VLAN is enabled after any MAC-Based VLAN sessions are established, all sessions are flushed and the mac-vlan-permit ports are removed from the VLAN. The ports are then added back to the VLAN dynamically after they successfully pass the RADIUS authentication process.

Configuration Notes

The following guidelines apply to MAC-Based VLAN configurations:

- MAC-to-VLAN mapping must be associated with VLANs that exist on the switch. Create the VLANs before you configure the MAC-Based VLAN feature.
- Ports participating in MAC-based VLANs must first be configured as **mac-vlan-permit** ports under the VLAN's configuration.
- In the RADIUS server configuration file, a MAC address cannot be configured to associate with more than one VLAN.
- This feature does not currently support dynamic assignment of a port to a VLAN. Users must pre-configure VLANs and port membership before enabling the feature.
- Multi-device port authentication filters will not work with MAC-Based VLANs on the same port.
- MAC-Based VLAN is not currently supported for trunk ports and LACP.
- MAC-Based VLAN is not supported for VLAN groups, topology groups and dual-mode configuration.

The following table describes the CLI commands used to configure MAC-Based VLANs.

Table 15.3: CLI Commands for MAC-Based VLANs

CLI Command	Description	CLI Level
mac-auth mac-vlan enable	Enables per-port MAC-Based VLAN	Interface
mac-auth mac-vlan disable	Disables per-port MAC-Based VLAN	interface
mac-auth mac-vlan-dyn-activation	Enables Dynamic MAC-Based VLAN	global
no mac-auth mac-vlan-dyn-activation	Disables Dynamic MAC-Based VLAN	global
no mac-auth mac-vlan	Removes the MAC-VLAN configuration from the port	interface
mac-auth mac-vlan max-mac-entries <num of entries>	The maximum number of allowed and denied MAC addresses (static and dynamic) that can be learned on a port. The default is 2.	interface
mac-auth mac-vlan <mac-addr> vlan <vlan id> priority <0-7>	Adds a static MAC-VLAN mapping to the MAC-Based VLAN table (for static hosts)	interface
clear table-mac-vlan	Clears the contents of the authenticated MAC address table	global
clear table-mac-vlan ethernet <port>	Clears all MAC-Based VLAN mapping on a port	global
show table-mac-vlan	Displays information about allowed and denied MAC addresses on ports with MAC-Based VLAN enabled.	global
show table-mac-vlan allowed-mac	Displays MAC addresses that have been successfully authenticated	global
show table-mac-vlan denied-mac	Displays MAC addresses for which authentication failed	global
show table-mac-vlan detailed	Displays detailed MAC-VLAN settings and classified MAC addresses for a port with the feature enabled	global
show table-mac-vlan <mac-address>	Displays status and details for a specific MAC address	global
show table-mac-vlan ethernet <port>	Displays all MAC addresses allowed or denied on a specific port	global

Configuration Example

The following example shows a MAC-Based VLAN configuration:

```
FLS624 Switch#show run
Current configuration:
ver 04.0.00b122T7e1
fan-threshold mp speed-3 35 100
module 1 fls-24-port-copper-base-module
module 4 fls-xfp-1-port-10g-module
vlan 1 by port
 untagged ethe 0/1/10
 mac-vlan-permit ethe 0/1/1 to 0/1/3
 no spanning-tree
vlan 2 by port
 untagged ethe 0/1/24
 mac-vlan-permit ethe 0/1/1 to 0/1/3
 no spanning-tree
vlan 222 name RESTRICTED_MBV by port
 untagged ethe 0/1/4
 mac-vlan-permit ethe 0/1/1 to 0/1/3
vlan 666 name RESTRICTED_MAC_AUTH by port
 untagged ethe 0/1/20
 mac-vlan-permit ethe 0/1/1 to 0/1/3
 spanning-tree 802-1w
vlan 4000 name DEFAULT-VLAN by port
vlan 4004 by port
 mac-vlan-permit ethe 0/1/1 ethe 0/1/3
default-vlan-id 4000
ip address 10.44.3.3 255.255.255.0
ip default-gateway 10.44.3.1
radius-server host 10.44.3.111
radius-server key 1 $-ndUno
mac-authentication enable
mac-authentication mac-vlan-dyn-activation
mac-authentication max-age 60
mac-authentication hw-deny-age 30
mac-authentication auth-passwd-format xxxx.xxxx.xxxx
mac-authentication auth-fail-vlan-id 666
interface ethernet 0/1/1
 mac-authentication mac-vlan max-mac-entries 5
 mac-authentication mac-vlan 0030.4888.b9fe vlan 1 priority 1
 mac-authentication mac-vlan enable
interface ethernet 0/1/2
 mac-authentication mac-vlan max-mac-entries 10
 mac-authentication mac-vlan enable
 mac-authentication auth-fail-action restrict-vlan 222
interface ethernet 0/1/3
 mac-authentication mac-vlan enable
 mac-authentication auth-fail-action restrict-vlan
!
end
```

Configuring MAC-Based VLANs Using SNMP

Platform Support:

- FGS and FLS devices running software release 04.0.00 and later

Several MIB objects have been developed to allow the configuration of MAC-Based VLANs using SNMP. For more information, refer to the September, 2007 *Foundry Management Information Base Reference*, in the MAC-Based VLAN chapter.

Displaying the MAC-VLAN Table

Enter the following command to display the MAC-VLAN table:

```
FGS648P Switch#show table-mac-vlan
----------------------------------------------------------------
Port    Vlan  Accepted  Rejected  Attempted  Static  Static  Max
              Macs      Macs      Macs       Macs    Conf    Macs
----------------------------------------------------------------
0/1/1    N/A  32        0         0          0       0       32
0/1/2    N/A  0         0         209        0       3       32
0/1/3    N/A  32        0         0          0       0       32
0/1/4    N/A  32        0         0          0       0       32
0/1/5    N/A  32        0         0          0       0       32
0/1/6    N/A  32        0         0          0       0       32
0/1/7    N/A  32        0         0          0       0       32
```

Syntax: show table-mac-vlan

The following table describes the information in this output:

This Field...	Displays...
Port	The port number where MAC-Based VLAN is enabled.
Vlan	Not applicable for this feature, will always display n/a.
Accepted Macs	The number of MAC addresses that have been successfully authenticated (dynamic hosts) combined with the number of active static MAC addresses (static hosts).
Rejected Macs	The number of MAC addresses for which authentication has failed for dynamic hosts.
Attempted Macs	The number of attempts made to authenticate MAC addresses.
Static Macs	The number of currently connected active static hosts.
Static Conf	The number of static hosts that are configured on the physical port.
Max Macs	The maximum number of allowed MAC addresses.

Displaying the MAC-VLAN Table for a Specific MAC Address

Enter the following command to display the MAC-VLAN table information for a specific MAC address:

```
FGS624#show table-mac-vlan 0030.4875.3f73
-------------------------------------------------------------------------------
MAC Address    Port    Vlan Authenticated  Time Age   CAM   MAC  Dot1x Type Pri
                                                      Index Index
-------------------------------------------------------------------------------
0030.4875.3f73 0/1/1   2    Yes    01d00h04m27s S0     0001  3728 Ena   Dyn  4
```

The following example shows output from the show table-mac-vlan command while authentication is in progress:

```
FGS624#show table-mac-vlan 0000.0200.001b
-------------------------------------------------------------------------------
MAC Address    Port  Vlan Authenticated  Time     Age  CAM   MAC  Dot1x Type Pri
                                                       Index Index
-------------------------------------------------------------------------------
0000.0200.001b 0/1/2 4092   Inp      00d00h00m00s S36   N/A  N/A   Dis   Dyn  0
```

The following table describes the information in these output examples:

This Field...	Displays...
MAC Address	The MAC address for which this information is displayed.
Port	The port where MAC-Based VLAN is enabled.
Vlan	The VLAN to which the MAC address has been assigned.
Authenticated	Yes indicates authentication is successful. No indicates authentication has failed. Inp indicates authentication in progress Rst indicates a restricted VLAN
Time	The time at which the MAC address was authenticated. If the clock is set on the Foundry device, then the actual date and time are displayed. If the clock has not been set, then the time is displayed relative to when the device was last restarted.
Age	The age of the MAC address entry in the authenticated MAC address list.
CAM Index	This field displays the index of the CAM entry. The index value will be between 0 and 31. A value of “ff” indicates that the index is not used.
MAC Index	The index of the entry in the hardware MAC table.
Dot1x	Indicates if 802.1X authentication is enabled or disabled for the MAC address.
Type	Dyn indicates a dynamic host. Sta indicates a static host.
Pri	This field indicates the value set for the Foundry-MAC-Based VLAN-QoS attribute (if configured) in the RADIUS configuration for dynamic hosts. If the Foundry-MAC-Based VLAN-QoS attribute is not configured, the value will be zero. For static hosts, the user-configured priority value for the MAC address is displayed.

Displaying Allowed MAC Addresses

Enter the following command to display information about successfully authenticated MAC addresses:

```
FGS648P Switch#show table-mac-vlan allowed-mac
-------------------------------------------------------------------------------
MAC Address    Port    Vlan Authenticated  Time Age   CAM   MAC  Dot1x Type Pri
                                                      Index Index
-------------------------------------------------------------------------------
0000.0100.0001 0/1/1   1    Yes     00d19h38m29s S60   0008  4000 Dis   Dyn  0
0000.0100.0002 0/1/1   1    Yes     00d19h38m29s S56   0009  4000 Dis   Dyn  1
0000.0100.0003 0/1/1   1    Yes     00d19h38m30s Ena   000a  2b44 Dis   Dyn  2
0000.0100.0004 0/1/1   1    Yes     00d19h38m49s Ena   0013  0c20 Dis   Dyn  3
```

Syntax: show table-mac-vlan allowed-mac

The following table describes the information in this output:

This Field...	Displays...
MAC Address	The allowed MAC addresses for which the information is displayed.
Port	The port where MAC-Based VLAN is enabled.
Vlan	The VLAN to which the MAC address has been assigned.
Authenticated	Yes indicates authentication has been successful. Inp indicates authentication is in progress.
Time	The time at which each MAC address was authenticated. If the clock is set on the Foundry device, then the actual date and time are displayed. If the clock has not been set, then the time is displayed relative to when the device was last restarted.
Age	The age of the MAC address entry in the authenticated MAC address list.
CAM Index	This field displays the index of the CAM entry. The index value will be between 0 and 31. A value of “ff” indicates that the index is not used.
MAC Index	The index of the entry in the hardware MAC table.
Dot1x	Indicates whether 802.1X authentication is enabled or disabled for each MAC address.
Type	Dyn indicates a dynamic host.
Pri	This field indicates the value set for Foundry-MAC-Based VLAN-QoS attribute in the RADIUS configuration for dynamic hosts, if configured. If the Foundry-MAC-Based VLAN-QoS attribute is not configured, the value will be zero. For static hosts, the user-configured priority value for the MAC address is displayed.

Displaying Denied MAC Addresses

Enter the following command to display information about denied (authentication failed) MAC addresses:

```
FGS648P Switch#show table-mac-vlan denied-mac
-------------------------------------------------------------------------------
MAC Address    Port    Vlan Authenticated  Time Age    CAM   MAC  Dot1x Type Pri
                                                        Index Index
-------------------------------------------------------------------------------
0000.1111.0107 0/2/1   4092 No     00d20h12m30s H64    0000  37dc Dis   Dyn  0
```

Syntax: show table-mac-vlan denied-mac

The following table describes the information in this output:

This Field...	Displays...
MAC Address	The denied MAC address for which the information is displayed.
Port	The port where MAC-Based VLAN is enabled.
Vlan	This field displays VLAN 4092 for blocked hosts, or the restricted VLAN ID if it is configured on the port.
Authenticated	No indicates that authentication has failed. Inp indicates that authentication is in progress.
Time	The time at which authenticated failed.
Age	The age of the MAC address entry in the authenticated MAC address list.
CAM Index	This field displays the index of the CAM entry. The index value will be between 0 and 31. A value of “ff” indicates that the index is not used.
MAC Index	The index of the entry in the hardware MAC table.
Dot1x	Indicates whether 802.1X authentication is disabled (Dis) or enabled (Ena) for this MAC address.
Type	Dyn Indicates a dynamic host.
Pri	This field indicates the value set for Foundry-MAC-Based VLAN-QoS attribute in the RADIUS configuration for dynamic hosts, if configured. If the Foundry-MAC-Based VLAN-QoS attribute is not configured, the value will be zero. For static hosts, the user-configured priority value for the MAC address is displayed.

Displaying Detailed MAC-VLAN Data

Enter the following command to display a detailed version of MAC-VLAN information:

```
FGS648P Switch#show table-mac-vlan detailed  e 0/1/2
Port                               : 0/1/2
Dynamic-Vlan  Assignment           : Disabled
RADIUS failure action              : Block Traffic
   Failure restrict use dot1x      : No
Override-restrict-vlan             : Yes
Vlan                               : (MAC-PERMIT-VLAN )
Port Vlan State                    : DEFAULT
802.1x override Dynamic PVID       : NO
Original PVID                      : 1
DOS attack protection              : Disabled
Accepted Mac Addresses             : 32
Authentication in progress         : 0
Authentication attempts            : 54
RADIUS timeouts                    : 16817
Num of MAC entries in TCAM         : 32
Num of MAC entries in MAC          : 32
Aging of MAC-sessions              : Enabled
Port move-back vlan                : Port-configured-vlan
Max-Age of sw mac session          : 60 seconds
hw age for denied mac              : 30 seconds
MAC Filter  applied                : No
-------------------------------------------------------------------------------
MAC Address    RADIUS       Authenticated  Time Age    CAM   MAC  Dot1x Type Pri
                                                       Index Index
-------------------------------------------------------------------------------
0000.0200.0012 0.0.0.0      No     00d00h00m00s S12    N/A   N/A  Dis   Dyn  0
0000.0200.0017 0.0.0.0      No     00d00h00m00s S20    N/A   N/A  Dis   Dyn  0
0000.0200.0018 0.0.0.0      No     00d00h00m00s S20    N/A   N/A  Dis   Dyn  0
0000.0100.000a 10.44.3.111  Yes    00d19h38m30s Ena    000b  22d4 Dis   Dyn  5
0000.0200.0019 0.0.0.0      No     00d00h00m00s S20    N/A   N/A  Dis   Dyn  0
0000.0200.001a 0.0.0.0      No     00d00h00m00s S20    N/A   N/A  Dis   Dyn  0
0000.0200.001b 0.0.0.0      No     00d00h00m00s S20    N/A   N/A  Dis   Dyn  0
0000.0200.001c 0.0.0.0      No     00d00h00m00s S20    N/A   N/A  Dis   Dyn  0
0000.0200.001d 0.0.0.0      No     00d00h00m00s S20    N/A   N/A  Dis   Dyn  0
-------------------------------------------------------------------------------
```

Displaying MAC-VLAN Information for a Specific Interface

Enter the following command to display MAC-VLAN information for a specific interface:

```
FGS648P Switch#show table-mac-vlan e 0/1/1
--------------------------------------------------------------------------------
MAC Address    Port    Vlan Authenticated  Time Age    CAM   MAC  Dot1x Type Pri
                                                       Index Index
--------------------------------------------------------------------------------
0000.0100.0001 0/1/1   1    Yes     00d19h38m29s Ena   0008  0970 Dis   Dyn  0
0000.0100.0002 0/1/1   1    Yes     00d19h38m29s Ena   0009  0a40 Dis   Dyn  1
0000.0100.0003 0/1/1   1    Yes     00d19h38m30s Ena   000a  2b44 Dis   Dyn  2
0000.0100.0004 0/1/1   1    Yes     00d19h38m49s S96   0013  4000 Dis   Dyn  3
0000.0100.0005 0/1/1   1    Yes     00d19h38m53s Ena   0014  2d24 Dis   Dyn  4
0000.0100.0006 0/1/1   1    Yes     00d19h38m53s Ena   0015  2e14 Dis   Dyn  5
0000.0100.0007 0/1/1   1    Yes     00d19h38m41s S80   000f  4000 Dis   Dyn  6
0000.0100.0008 0/1/1   1    Yes     00d19h39m07s Ena   001f  00e0 Dis   Dyn  7
0000.0100.000a 0/1/1   1    Yes     00d19h38m30s Ena   000b  22d4 Dis   Dyn  0
0000.0100.0009 0/1/1   1    Yes     00d19h38m19s Ena   0001  21e4 Dis   Dyn  0
0000.0100.000a 0/1/1   1    Yes     00d19h38m30s Ena   000b  22d4 Dis   Dyn  0
0000.0100.000b 0/1/1   1    Yes     00d19h38m19s Ena   0002  03d0 Dis   Dyn  0
0000.0100.000c 0/1/1   1    Yes     00d19h38m57s Ena   001a  24b4 Dis   Dyn  0
0000.0100.000d 0/1/1   1    Yes     00d19h38m19s Ena   0003  05b0 Dis   Dyn  0
0000.0100.000e 0/1/1   1    Yes     00d19h38m31s S120  000c  4000 Dis   Dyn  0
```

The following table describes the information in this output:

This Field...	Displays...
MAC Address	The MAC addresses related to the specified interface.
Port	The interface for which this information is displayed.
Vlan	The VLAN to which the interface has been assigned.
Authenticated	Yes indicates authentication is successful. No indicates authentication has failed. Inp indicates authentication in progress Rst indicates a restricted VLAN
Time	The time at which the MAC address was authenticated. If the clock is set on the Foundry device, then the actual date and time are displayed. If the clock has not been set, then the time is displayed relative to when the device was last restarted.
Age	The age of the MAC address entry in the authenticated MAC address list.
CAM Index	This field displays the index of the CAM entry. The index value will be between 0 and 31. A value of “ff” indicates that the index is not used.
MAC Index	The index of the entry in the hardware MAC table.
Dot1x	Indicates whether 802.1X authentication is enabled or disabled for this MAC address.
Type	Dyn Indicates a dynamic host. Sta indicates a static host.
Pri	This field indicates the value set for Foundry-MAC-Based VLAN-QoS attribute in the RADIUS configuration for dynamic hosts, if configured. If the Foundry-MAC-Based VLAN-QoS attribute is not configured, the value will be zero. For static hosts, the user-configured priority value for the MAC address is displayed.

Clearing MAC-VLAN Information for a Specific Interface

Enter the following command to clear MAC-VLAN information. Add the interface id to clear information for a specific interface:

```
FGS648P Switch#clear table-mac-vlan <interface>
```

Displaying MAC addresses in a MAC-Based VLAN

Enter the following command to display a list of MAC addresses in a MAC-Based VLAN:

```
FGS648P Switch#show mac-address
Total active entries from all ports = 1541
MAC-Address     Port             Type          Index      VLAN
0000.2000.0001  0/1/32           Dynamic(MBV)  1048       1
0000.2000.0002  0/1/32           Dynamic(MBV)  1832       1
0000.2000.0003  0/1/32           Dynamic(MBV)  9772       1
0000.2000.0004  0/1/32           Static(MBV)   328        1
0000.2000.0005  0/1/32           Dynamic(MBV)  8268       1
0000.2000.0006  0/1/32           Dynamic(MBV)  9084       1
0000.2000.0007  0/1/32           Dynamic(MBV)  632        1
0000.2000.0008  0/1/32           Dynamic(MBV)  3464       1
0000.2000.0009  0/1/32           Dynamic(MBV)  11404      1
0000.2000.000a  0/1/32           Dynamic(MBV)  12220      1
0000.2000.000b  0/1/32           Dynamic(MBV)  3768       1
```

NOTE: In this output, (MBV) indicates MAC-Based VLAN is enabled.

The following table describes the output from this command:

This Field...	Displays...
Total active entries	The total number of active entries for all ports.
MAC Address	The MAC addresses assigned to this VLAN.
Port	The interface for which this information is displayed.
Type	Dynamic (MBV) Indicates a dynamic host. Static (MBV) indicates a static host.
Index	The index of the entry in the hardware MAC table.
VLAN	The VLAN to which these addresses are assigned.

Displaying MAC-Based VLAN Logging

Enter the following command to display MAC-Based VLAN logging activity:

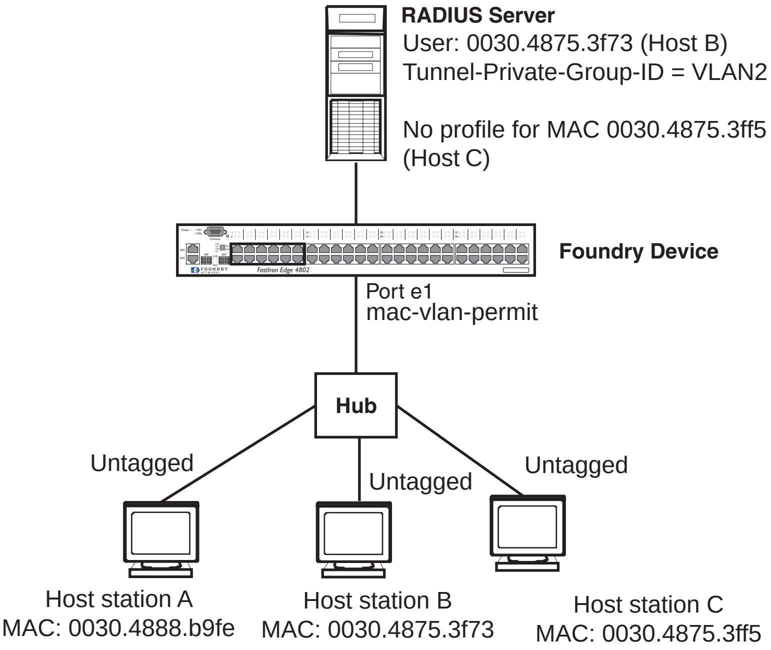
```
FGS648P Switch#show logging
Syslog logging: enabled (0 messages dropped, 0 flushes, 15 overruns)
    Buffer logging: level ACDMEINW, 50 messages logged
    level code: A=alert C=critical D=debugging M=emergency E=error
Static Log Buffer
0d00h00m12s:A:System:  Power supply 1  is up
Dynamic Log Buffer (50 lines):
0d18h46m28s:I:running-config was changed from console
0d02h12m25s:A:MAC Based Vlan Mapping failed for [0000.1111.0108 ] on port 0/2/1
(Invalid User)
0d02h08m52s:A:MAC Based Vlan Mapping failed for [0000.1111.011b ] on port 0/2/1
(Invalid User)
0d02h05m01s:A:MAC Based Vlan Mapping failed for [0000.1111.00df ] on port 0/2/1
(Invalid User)
0d02h01m15s:A:MAC Based Vlan Mapping failed for [0000.1111.0108 ] on port 0/2/1
(Invalid User)
0d02h01m15s:A:MAC Based Vlan Mapping failed for [0000.1111.0107 ] on port 0/2/1
(Invalid User)
0d01h58m43s:N:MAC Based Vlan Enabled on port 0/2/1
0d01h58m32s:N:MAC Based Vlan Disabled on port 0/2/1
0d01h39m00s:I:running-config was changed from console
0d01h38m28s:I:System: Interface ethernet 0/1/47, state up
0d01h38m27s:I:System: Interface ethernet 0/1/46, state up
0d01h38m27s:I:System: Interface ethernet 0/1/34, state up
0d01h38m27s:I:System: Interface ethernet 0/1/25, state up
```

Sample Application

Figure 15.1 illustrates a sample configuration that uses MAC-Based VLAN on port e 0/1/1 on the Foundry device. In this configuration, three host PCs are connected to port e 0/1/1via a hub.

Host A’s MAC address is statically configured on port e 0/1/1. The profile for Host B’s MAC address on the RADIUS server specifies that the PC should be assigned to VLAN 2. Host C’s profile does not exist in the RADIUS server, and will be put into a restricted VLAN.

Figure 15.1 Sample MAC-Based VLAN Configuration



Host A’s MAC address is statically mapped to VLAN 1 with priority 1 and is not subjected to RADIUS authentication. When Host B’s MAC address is authenticated, the Access-Accept message from the RADIUS server specifies that Host B’s MAC address be placed into VLAN 2. Since Host C’s MAC address is not present in the RADIUS server, Host C will be rejected by the server and its MAC address will be placed into a restricted VLAN.

Below is the configuration for this example:

```
module 1 fgs-48-port-management-module
module 2 fgs-xfp-1-cx4-1-port-10g-module
vlan 1 by port
 untagged ethe 0/1/10
 mac-vlan-permit ethe 0/1/1 to 0/1/2
 no spanning-tree
vlan 2 by port
 untagged ethe 0/1/30
 mac-vlan-permit ethe 0/1/1 to 0/1/2
 no spanning-tree
vlan 666 name mac_restricted by port
 untagged ethe 0/1/20
 mac-vlan-permit ethe 0/1/1 to 0/1/2
 no spanning-tree
vlan 4000 name DEFAULT-VLAN by port
 no spanning-tree
vlan 4004 by port
```

```
 mac-vlan-permit ethe 0/1/1
default-vlan-id 4000
ip address 10.44.3.8 255.255.255.0
ip default-gateway 10.44.3.1
radius-server host 10.44.3.111
radius-server key 1 $-ndUno
mac-authentication enable
mac-authentication max-age 60
mac-authentication hw-deny-age 30
mac-authentication auth-passwd-format xxxx.xxxx.xxxx
interface ethernet 0/1/1
 mac-authentication mac-vlan max-mac-entries 5
 mac-authentication mac-vlan 0030.4888.b9fe vlan 1 priority 1
 mac-authentication mac-vlan enable
!
interface ethernet 0/1/2
mac-authentication mac-vlan max-mac-entries 5
mac-authentication mac-vlan enable
!
!
end
```

The **show table-mac-vlan** command returns the following results for all ports in this configuration:

```
FGS648P Switch#show table-mac-vlan

---------------------------------------------------------------
Port   Vlan  Accepted  Rejected  Attempted  Static  Static  Max
             Macs      Macs      Macs       Macs    Conf    Macs
----------------------------------------------------------------
0/1/1   N/A  2         1         0          1       1       5
0/1/2   N/A  0         0         0          0       0       5
```

The **show table-mac-vlan e 0/1/1** command returns the following results for port 0/1/1 in this configuration:

```
FGS648P Switch#show table-mac-vlan  e 0/1/1

-------------------------------------------------------------------------------
MAC Address    Port    Vlan Authenticated  Time Age   CAM   MAC  Dot1x Type Pri
                                                      Index Index
-------------------------------------------------------------------------------
0030.4875.3f73 0/1/1   2    Yes    00d00h00m46s S32   0001  3728 Dis   Dyn  4
0030.4888.b9fe 0/1/1   1    Yes    00d00h00m08s Dis   0000  0970 Dis   Sta  1
0030.4875.3ff5 0/1/1   666  Rst    01d18h47m58s S8    0002  1ee4 Dis   Dyn  0
```

Chapter 16

Configuring GARP VLAN Registration Protocol (GVRP)

This chapter describes how to configure the GARP VLAN Registration Protocol (GVRP).

GVRP Overview

GARP VLAN Registration Protocol (GVRP) is a Generic Attribute Registration Protocol (GARP) application that provides VLAN registration service by means of dynamic configuration (registration) and distribution of VLAN membership information.

A Foundry device enabled for GVRP can do the following:

- Learn about VLANs from other Foundry devices and configure those VLANs on the ports that learn about the VLANs. The device listens for GVRP Protocol Data Units (PDUs) from other devices, and implements the VLAN configuration information in the PDUs.
- Advertise VLANs configured on the device to other Foundry devices. The device sends GVRP PDUs advertising its VLANs to other devices. GVRP advertises statically configured VLANs and VLANs learned from other devices through GVRP.

GVRP enables a Foundry device to dynamically create 802.1Q-compliant VLANs on links with other devices that are running GVRP. GVRP reduces the chances for errors in VLAN configuration by automatically providing VLAN ID consistency across the network. You can use GVRP to propagate VLANs to other GVRP-aware devices automatically, without the need to manually configure the VLANs on each device. In addition, if the VLAN configuration on a device changes, GVRP automatically changes the VLAN configurations of the affected devices.

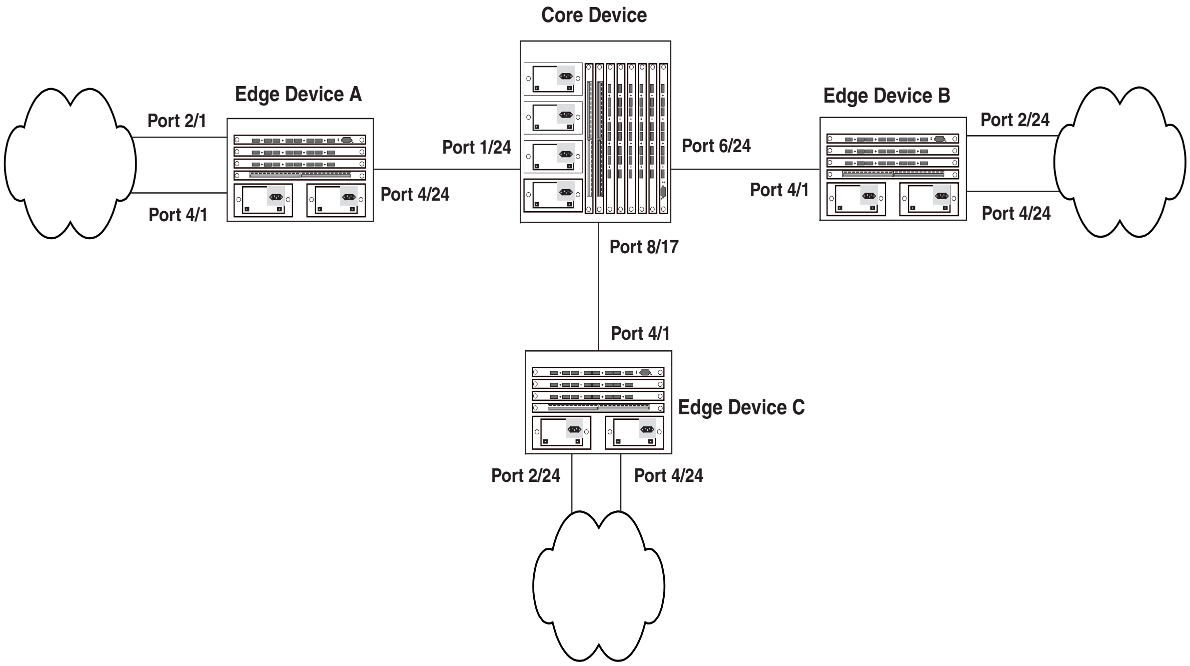
The Foundry implementation of GARP and GVRP is based on the following standards:

- ANSI/IEEE standard 802.1D, 1998 edition
- IEEE standard 802.1Q, 1998 edition; approved December 8, 1998
- IEEE draft P802.1w/D10, March 26, 2001
- IEEE draft P802.1u/D9, November 23, 2000
- IEEE draft P802.1t/D10, November 20, 2000

Application Examples

Figure 16.1 shows an example of a network that uses GVRP. This section describes various ways you can use GVRP in a network such as this one. “CLI Examples” on page 16-17 lists the CLI commands to implement the applications of GVRP described in this section.

Figure 16.1 Example of GVRP



In this example, a core device is attached to three edge devices. Each of the edge devices is attached to other edge devices or host stations (represented by the clouds).

The effects of GVRP in this network depend on which devices the feature is enabled on, and whether both learning and advertising are enabled. In this type of network (a core device and edge devices), you can have the following four combinations:

- Dynamic core and fixed edge
- Dynamic core and dynamic edge
- Fixed core and dynamic edge
- Fixed core and fixed edge

Dynamic Core and Fixed Edge

In this configuration, all ports on the core device are enabled to learn and advertise VLAN information. The edge devices are configured to advertise their VLAN configurations on the ports connected to the core device. GVRP learning is disabled on the edge devices.

Core Device	Edge Device A	Edge Device B	Edge Device C
GVRP is enabled on all ports. Both learning and advertising are enabled. **Note**: Since learning is disabled on all the edge devices, advertising on the core device has no effect in this configuration.	GVRP is enabled on port 4/24. Learning is disabled. VLAN 20 Port 2/1 (untagged) Port 4/24 (tagged) VLAN 40 Port 4/1 (untagged) Port 4/24 (tagged)	GVRP is enabled on port 4/1. Learning is disabled. VLAN 20 Port 2/24 (untagged) Port 4/1 (tagged) VLAN 30 Port 4/24 (untagged) Port 4/1 (tagged)	GVRP is enabled on port 4/1. Learning is disabled. VLAN 30 Port 2/24 (untagged) Port 4/1 (tagged) VLAN 40 Port 4/24 (untagged) Port 4/1 (tagged)

In this configuration, the edge devices are statically (manually) configured with VLAN information. The core device dynamically configures itself to be a member of each of the edge device's VLANs. The operation of GVRP on the core device results in the following VLAN configuration on the device:

- VLAN 20
  - 1/24 (tagged)
  - 6/24 (tagged)
- VLAN 30
  - 6/24 (tagged)
  - 8/17 (tagged)
- VLAN 40
  - 1/24 (tagged)
  - 8/17 (tagged)

VLAN 20 traffic can now travel through the core between edge devices A and B. Likewise, VLAN 30 traffic can travel between B and C and VLAN 40 traffic can travel between A and C. If an edge device is moved to a different core port or the VLAN configuration of an edge device is changed, the core device automatically reconfigures itself to accommodate the change.

Notice that each of the ports in the dynamically created VLANs is tagged. All GVRP VLAN ports configured by GVRP are tagged, to ensure that the port can be configured for additional VLANs.

NOTE: This example assumes that the core device has no static VLANs configured. However, you can have static VLANs on a device that is running GVRP. GVRP can dynamically add other ports to the statically configured VLANs but cannot delete statically configured ports from the VLANs.

Dynamic Core and Dynamic Edge

GVRP is enabled on the core device and on the edge devices. This type of configuration is useful if the devices in the edge clouds are running GVRP and advertise their VLANs to the edge devices. The edge devices learn the VLANs and also advertise them to the core. In this configuration, you do not need to statically configure the

VLANs on the edge or core devices, although you can have statically configured VLANs on the devices. The devices learn the VLANs from the devices in the edge clouds.

Fixed Core and Dynamic Edge

GVRP learning is enabled on the edge devices. The VLANs on the core device are statically configured, and the core device is enabled to advertise its VLANs but not to learn VLANs. The edge devices learn the VLANs from the core.

Fixed Core and Fixed Edge

The VLANs are statically configured on the core and edge devices. On each edge device, VLAN advertising is enabled but learning is disabled. GVRP is not enabled on the core device. This configuration enables the devices in the edge clouds to learn the VLANs configured on the edge devices.

VLAN Names

The **show vlans** command lists VLANs created by GVRP as “GVRP_VLAN_<vlan-id>”. VLAN names for statically configured VLANs are not affected. To distinguish between statically-configured VLANs that you add to the device and VLANs that you convert from GVRP-configured VLANs into statically-configured VLANs, the **show vlans** command displays a converted VLAN’s name as “STATIC_VLAN_<vlan-id>”.

Configuration Notes

- If you disable GVRP, all GVRP configuration information is lost if you save the configuration change (**write memory** command) and then reload the software. However, if you reload the software without first saving the configuration change, the GVRP configuration is restored following a software reload.
- The maximum number of VLANS supported on a device enabled for GVRP is the same as the maximum number on a device that is not enabled for GVRP.
  - To display the maximum number of VLANs allowed on your device, enter the **show default values** command. See the “vlan” row in the System Parameters section. Make sure you allow for the default VLAN (1), the GVRP base VLAN (4093), and the Single STP VLAN (4094). These VLANs are maintained as “Registration Forbidden” in the GVRP database. Registration Forbidden VLANs cannot be advertised or learned by GVRP.
  - To increase the maximum number of VLANs supported on the device, enter the **system-max vlan** <num> command at the global CONFIG level of the CLI, then save the configuration and reload the software. The maximum number you can specify is listed in the Maximum column of the **show default values** display.
- The default VLAN (VLAN 1) is not advertised by the Foundry implementation of GVRP. The default VLAN contains all ports that are not members of statically configured VLANs or VLANs enabled for GVRP.

  NOTE: The default VLAN has ID 1 by default. You can change the VLAN ID of the default VLAN, but only before GVRP is enabled. You cannot change the ID of the default VLAN after GVRP is enabled.

- Single STP must be enabled on the device. Foundry’s implementation of GVRP requires Single STP. If you do not have any statically configured VLANs on the device, you can enable Single STP as follows:

  ```
  FastIron(config)#vlan 1
  FastIron(config-vlan-1)#exit
  FastIron(config)#span
  FastIron(config)#span single
  ```

  These commands enable configuration of the default VLAN (VLAN 1), which contains all the device’s ports, and enable STP and Single STP.

- All VLANs that are learned dynamically through GVRP are added to the single spanning tree.

- All ports that are enabled for GVRP become tagged members of the GVRP base VLAN (4093). If you need to use this VLAN ID for another VLAN, you can change the GVRP VLAN ID. See "Changing the GVRP Base VLAN ID" on page 16-5. The software adds the GVRP base VLAN to the single spanning tree.
- All VLAN ports added by GVRP are tagged.
- GVRP is supported only for tagged ports or for untagged ports that are members of the default VLAN. GVRP is not supported for ports that are untagged and are members of a VLAN other than the default VLAN.
- To configure GVRP on a trunk group, enable the protocol on the primary port in the trunk group. The GVRP configuration of the primary port is automatically applied to the other ports in the trunk group.
- You can use GVRP on a device even if the device has statically configured VLANs. GVRP does not remove any ports from the statically configured VLANs, although GVRP can add ports to the VLANS. GVRP advertises the statically configured VLANs. Ports added by GVRP do not appear in the running-config and will not appear in the startup-config file when save the configuration. You can manually add a port to make the port a permanent member of the VLAN. After you manually add the port, the port will appear in the running-config and be saved to the startup-config file when you save the configuration.
- VLANs created by GVRP do not support virtual routing interfaces or protocol-based VLANs. virtual routing interfaces and protocol-based VLANs are still supported on statically configured VLANs even if GVRP adds ports to those VLANs.
- You cannot manually configure any parameters on a VLAN that is created by GVRP. For example, you cannot change STP parameters for the VLAN.
- The GVRP timers (Join, Leave, and Leaveall) must be set to the same values on all the devices that are exchanging information using GVRP.
- If the network has a large number of VLANs, the GVRP traffic can use a lot of CPU resources. If you notice high CPU utilization after enabling GVRP, set the GVRP timers to longer values. In particular, set the Leaveall timer to a longer value. See "Changing the GVRP Timers" on page 16-7.
- The feature is supported only on Ethernet ports.

NOTE: If you plan to change the GVRP base VLAN ID (4093) or the maximum configurable value for the Leaveall timer (300000 ms by default), you must do so before you enable GVRP.

Configuring GVRP

To configure a device for GVRP, globally enable support for the feature, then enable the feature on specific ports. Optionally, you can disable VLAN learning or advertising on specific interfaces.

You also can change the protocol timers and change the GVRP base VLAN ID.

Changing the GVRP Base VLAN ID

By default, GVRP uses VLAN 4093 as a base VLAN for the protocol. All ports that are enabled for GVRP become tagged members of this VLAN. If you need to use VLAN ID 4093 for a statically configured VLAN, you can change the GVRP base VLAN ID.

NOTE: If you want to change the GVRP base VLAN ID, you must do so before enabling GVRP.

To change the GVRP base VLAN ID, enter a command such as the following at the global CONFIG level of the CLI:

```
FastIron(config)#gvrp-base-vlan-id 1001
```

This command changes the GVRP VLAN ID from 4093 to 1001.

Syntax: [no] gvrp-base-vlan-id <vlan-id>

The <vlan-id> parameter specifies the new VLAN ID. You can specify a VLAN ID from 2 – 4092 or 4095.

Increasing the Maximum Configurable Value of the Leaveall Timer

By default, the highest value you can specify for the Leaveall timer is 300000 ms. You can increase the maximum configurable value of the Leaveall timer to 1000000 ms.

NOTE: You must enter this command before enabling GVRP. Once GVRP is enabled, you cannot change the maximum Leaveall timer value.

NOTE: This command does not change the default value of the Leaveall timer itself. The command only changes the maximum value to which you can set the Leaveall timer.

To increase the maximum value you can specify for the Leaveall timer, enter a command such as the following at the global CONFIG level of the CLI:

```
FastIron(config)#gvrp-max-leaveall-timer 1000000
```

Syntax: [no] gvrp-max-leaveall-timer <ms>

The <ms> parameter specifies the maximum number of ms to which you can set the Leaveall timer. You can specify from 300000 – 1000000 (one million) ms. The value must be a multiple of 100 ms. The default is 300000 ms.

Enabling GVRP

To enable GVRP, enter commands such as the following at the global CONFIG level of the CLI:

```
FastIron(config)#gvrp-enable
FastIron(config-gvrp)#enable all
```

The first command globally enables support for the feature and changes the CLI to the GVRP configuration level. The second command enables GVRP on all ports on the device.

The following command enables GVRP on ports 1/24, 2/24, and 4/17:

```
FastIron(config-gvrp)#enable ethernet 1/24 ethernet 2/24 ethernet 4/17
```

Syntax: [no] gvrp-enable

Syntax: [no] enable all | ethernet [<slotnum>/] <portnum> [ethernet [<slotnum>/] <portnum> | to [<slotnum>/] <portnum>]

The **all** parameter enables GVRP on all ports.

The **ethernet** [<slotnum>/] <portnum> [**ethernet** [<slotnum>/] <portnum> | **to** [<slotnum>/] <portnum>] parameter enables GVRP on the specified list or range of Ethernet ports.

- To specify a list, enter each port as **ethernet** <portnum> followed by a space. Include the slot number if required. For example, to enable GVRP on three Ethernet ports on a FSX device, enter the following command: **enable ethernet 1/24 ethernet 6/24 ethernet 8/17**
- To specify a range, enter the first port in the range as **ethernet** [<slotnum>/] <portnum> followed by **to** followed by the last port in the range. For example, to add ports 1/1 – 1/8, enter the following command: **enable ethernet 1/1 to 1/8**.

You can combine lists and ranges in the same command. For example: **enable ethernet 1/1 to 1/8 ethernet 1/24 ethernet 6/24 ethernet 8/17**

Disabling VLAN Advertising

To disable VLAN advertising on a port enabled for GVRP, enter a command such as the following at the GVRP configuration level:

```
FastIron(config-gvrp)#block-applicant ethernet 1/24 ethernet 6/24 ethernet 8/17
```

This command disables advertising of VLAN information on ports 1/24, 6/24, and 8/17.

Syntax: [no] block-applicant all | ethernet [<slotnum>/] <portnum> [ethernet [<slotnum>/] <portnum> | to [<slotnum>/] <portnum>]

NOTE: Leaveall messages are still sent on the GVRP ports.

Disabling VLAN Learning

To disable VLAN learning on a port enabled for GVRP, enter a command such as the following at the GVRP configuration level:

```
FastIron(config-gvrp)#block-learning ethernet 6/24
```

This command disables learning of VLAN information on port 6/24.

NOTE: The port still advertises VLAN information unless you also disable VLAN advertising.

Syntax: [no] block-learning all | ethernet [<slotnum>/] <portnum> [ethernet [<slotnum>/] <portnum> | to [<slotnum>/] <portnum>]

Changing the GVRP Timers

GVRP uses the following timers:

- Join – The maximum number of milliseconds (ms) a device’s GVRP interfaces wait before sending VLAN advertisements on the interfaces. The actual interval between Join messages is randomly calculated to a value between 0 and the maximum number of milliseconds specified for Join messages. You can set the Join timer to a value from 200 – one third the value of the Leave timer. The default is 200 ms.
- Leave – The number of ms a GVRP interface waits after receiving a Leave message on the port to remove the port from the VLAN indicated in the Leave message. If the port receives a Join message before the Leave timer expires, GVRP keeps the port in the VLAN. Otherwise, the port is removed from the VLAN. When a port receives a Leave message, the port’s GVRP state is changed to Leaving. Once the Leave timer expires, the port’s GVRP state changes to Empty. You can set the Leave timer to a value from three times the Join timer – one fifth the value of the Leaveall timer. The default is 600 ms.

  NOTE: When all ports in a dynamically created VLAN (one learned through GVRP) leave the VLAN, the VLAN is immediately deleted from the device's VLAN database. However, this empty VLAN is still maintained in the GVRP database for an amount of time equal to the following:

  (number-of-GVRP-enabled-up-ports) * (2 * join-timer)

  While the empty VLAN is in the GVRP database, the VLAN does not appear in the **show vlans** display but does still appear in the **show gvrp vlan all** display.

- Leaveall – The minimum interval at which GVRP sends Leaveall messages on all GVRP interfaces. Leaveall messages ensure that the GVRP VLAN membership information is current by aging out stale VLAN information and adding information for new VLAN memberships, if the information is missing. A Leaveall message instructs the port to change the GVRP state for all its VLANs to Leaving, and remove them unless a Join message is received before the Leave timer expires. By default, you can set the Leaveall timer to a value from five times the Leave timer – maximum value allowed by software (configurable from 300000 – 1000000 ms). The default is 10000.

  NOTE: The actual interval is a random value between the Leaveall interval and 1.5 * the Leaveall time or the maximum Leaveall time, whichever is lower.

NOTE: You can increase the maximum configurable value of the Leaveall timer from 300000 ms up to 1000000 ms using the **gvrp-max-leaveall-timer** command. (See “Increasing the Maximum Configurable Value of the Leaveall Timer” on page 16-6.)

Timer Configuration Requirements

- All timer values must be in multiples of 100 ms.
- The Leave timer must be >= 3* the Join timer.
- The Leaveall timer must be >= 5* the Leave timer.
- The GVRP timers must be set to the same values on all the devices that are exchanging information using GVRP.

Changing the Join, Leave, and Leaveall Timers

The same CLI command controls changes to the Join, Leave, and Leaveall timers. To change values to the timers, enter a command such as the following:

```
FastIron(config-gvrp)#join-timer 1000 leave-timer 3000 leaveall-timer 15000
```

This command changes the Join timer to 1000 ms, the Leave timer to 3000 ms, and the Leaveall timer to 15000.

Syntax: [no] join-timer <ms> leave-timer <ms> leaveall-timer <ms>

NOTE: When you enter this command, all the running GVRP timers are canceled and restarted using the new times specified by the command.

Resetting the Timers to Their Defaults

To reset the Join, Leave, and Leaveall timers to their default values, enter the following command:

```
FastIron(config-gvrp)#default-timers
```

Syntax: default-timers

This command resets the timers to the following values:

- Join – 200 ms
- Leave – 600 ms
- Leaveall – 10000 ms

Converting a VLAN Created by GVRP into a Statically-Configured VLAN

You cannot configure VLAN parameters on VLANs created by GVRP. Moreover, VLANs and VLAN ports added by GVRP do not appear in the running-config and cannot be saved in the startup-config file.

To be able to configure and save VLANs or ports added by GVRP, you must convert the VLAN ports to statically-configured ports.

To convert a VLAN added by GVRP into a statically-configured VLAN, add the ports using commands such as the following:

```
FastIron(config)#vlan 22
FastIron(config-vlan-222)#tagged ethernet 1/1 to 1/8
```

These commands convert GVRP-created VLAN 22 containing ports 1/1 through 1/8 into statically-configured VLAN 22.

Syntax: [no] vlan <vlan-id>

Syntax: [no] tagged ethernet [<slotnum>/] <portnum> [to [<slotnum>/] <portnum> | ethernet [<slotnum>/] <portnum>]

Use the same commands to statically add ports that GVRP added to a VLAN.

NOTE: You cannot add the VLAN ports as untagged ports.

NOTE: After you convert the VLAN, the VLAN name changes from "'GVRP_VLAN_<vlan-id>" to "STATIC_VLAN_<vlan-id>".

Displaying GVRP Information

You can display the following GVRP information:

- GVRP configuration information
- GVRP VLAN information
- GVRP statistics
- CPU utilization statistics
- GVRP diagnostic information

Displaying GVRP Configuration Information

To display GVRP configuration information, enter a command such as the following:

```
FastIron#show gvrp
GVRP is enabled on the system

GVRP BASE VLAN ID          : 4093
GVRP MAX Leaveall Timer    : 300000 ms

GVRP Join Timer            : 200 ms
GVRP Leave Timer           : 600 ms
GVRP Leave-all Timer       : 10000 ms

===========================================================================
Configuration that is being used:

 block-learning ethe 1/3
 block-applicant ethe 2/7 ethe 2/11
 enable ethe 1/1 to 1/7 ethe 2/1 ethe 2/7 ethe 2/11

===========================================================================

Spanning Tree: SINGLE SPANNING TREE
Dropped Packets Count: 0

===========================================================================

Number of VLANs in the GVRP Database: 15
Maximum Number of VLANs that can be present: 4095

===========================================================================
```

Syntax: show gvrp [ethernet <port-num>]

This display shows the following information.

Table 16.1: CLI Display of Summary GVRP Information

This Field...	Displays...
Protocol state	The state of GVRP. The display shows one of the following: • GVRP is disabled on the system • GVRP is enabled on the system
GVRP BASE VLAN ID	The ID of the base VLAN used by GVRP.
GVRP MAX Leaveall Timer	The maximum number of ms to which you can set the Leaveall timer. **Note**: To change the maximum value, see “Increasing the Maximum Configurable Value of the Leaveall Timer” on page 16-6.
GVRP Join Timer	The value of the Join timer. **Note**: For descriptions of the Join, Leave, and Leaveall timers or to change the timers, see “Changing the GVRP Timers” on page 16-7.
GVRP Leave Timer	The value of the Leave timer.
GVRP Leave-all Timer	The value of the Leaveall timer.
Configuration that is being used	The configuration commands used to enable GVRP on individual ports. If GVRP learning or advertising is disabled on a port, this information also is displayed.
Spanning Tree	The type of STP enabled on the device. **Note**: The current release supports GVRP only with Single STP.
Dropped Packets Count	The number of GVRP packets that the device has dropped. A GVRP packet can be dropped for either of the following reasons: • GVRP packets are received on a port on which GVRP is not enabled. **Note**: If GVRP support is not globally enabled, the device does not drop the GVRP packets but instead forwards them at Layer 2. • GVRP packets are received with an invalid GARP Protocol ID. The protocol ID must always be 0x0001.
Number of VLANs in the GVRP Database	The number of VLANs in the GVRP database. **Note**: This number includes the default VLAN (1), the GVRP base VLAN (4093), and the single STP VLAN (4094). These VLANs are not advertised by GVRP but are maintained as “Registration Forbidden”.
Maximum Number of VLANs that can be present	The maximum number of VLANs that can be configured on the device. This number includes statically configured VLANs, VLANs learned through GVRP, and VLANs 1, 4093, and 4094. To change the maximum number of VLANs the device can have, use the **system-max vlan** <num> command. See “Displaying and Modifying System Parameter Default Settings” on page 8-11.

To display detailed GVRP information for an individual port, enter a command such as the following:

```
FastIron#show gvrp ethernet 2/1
Port 2/1 -
 GVRP Enabled   : YES
 GVRP Learning  : ALLOWED
 GVRP Applicant : ALLOWED
 Port State     : UP
 Forwarding     : YES

 VLAN Membership:        [VLAN-ID]            [MODE]
                                 1         FORBIDDEN
                                 2             FIXED
                              1001            NORMAL
                              1003            NORMAL
                              1004            NORMAL
                              1007            NORMAL
                              1009            NORMAL
                              1501            NORMAL
                              2507            NORMAL
                              4001            NORMAL
                              4093         FORBIDDEN
                              4094         FORBIDDEN
```

This display shows the following information.

Table 16.2: CLI Display of Detailed GVRP Information for a Port

This Field...	Displays...
Port number	The port for which information is being displayed.
GVRP Enabled	Whether GVRP is enabled on the port.
GVRP Learning	Whether the port can learn VLAN information from GVRP.
GVRP Applicant	Whether the port can advertise VLAN information into GVRP.
Port State	The port’s link state, which can be UP or DOWN.
Forwarding	Whether the port is in the GVRP Forwarding state: • NO – The port is in the Blocking state. • YES – The port is in the Forwarding state.

Table 16.2: CLI Display of Detailed GVRP Information for a Port (Continued)

This Field...	Displays...
VLAN Membership	The VLANs of which the port is a member. For each VLAN, the following information is shown: • VLAN ID – The VLAN’s ID. • Mode – The type of VLAN, which can be one of the following: • FIXED – The port will always be a member of this VLAN and the VLAN will always be advertised on this port by GVRP. A port becomes FIXED when you configure the port as a tagged member of a statically configured VLAN. • FORBIDDEN – The VLAN is one of the special VLANs that is not advertised or learned by GVRP. In the current release, the following VLANs are forbidden: the default VLAN (1), the GVRP base VLAN (4093), or the Single STP VLAN (4094). • NORMAL – The port became a member of this VLAN after learning about the VLAN through GVRP. The port’s membership in the VLAN depends on GVRP. If the VLAN is removed from the ports that send GVRP advertisements to this device, then the port will stop being a member of the VLAN.

Displaying GVRP VLAN Information

To display information about all the VLANs on the device, enter the following command:

```
FastIron#show gvrp vlan brief

Number of VLANs in the GVRP Database: 7
Maximum Number of VLANs that can be present: 4095

        [VLAN-ID]                          [MODE]       [VLAN-INDEX]

                1                  STATIC-DEFAULT                  0
                7                          STATIC                  2
               11                          STATIC                  4
             1001                         DYNAMIC                  7
             1003                         DYNAMIC                  8
             4093           STATIC-GVRP-BASE-VLAN                  6
             4094         STATIC-SINGLE-SPAN-VLAN                  5

===========================================================================
```

Syntax: show gvrp vlan all | brief | <vlan-id>

This display shows the following information.

Table 16.3: CLI Display of Summary VLAN Information for GVRP

This Field...	Displays...
Number of VLANs in the GVRP Database	The number of VLANs in the GVRP database. **Note**: This number includes the default VLAN (1), the GVRP base VLAN (4093), and the single STP VLAN (4094). These VLANs are not advertised by GVRP but are included in the total count.
Maximum Number of VLANs that can be present	The maximum number of VLANs that can be configured on the device. This number includes statically configured VLANs, VLANs learned through GVRP, and VLANs 1, 4093, and 4094. To change the maximum number of VLANs the device can have, use the **system-max vlan** <num> command. See "Displaying and Modifying System Parameter Default Settings" on page 8-11.
VLAN-ID	The VLAN ID.
MODE	The type of VLAN, which can be one of the following: • STATIC – The VLAN is statically configured and cannot be removed by GVRP. This includes VLANs you have configured as well as the default VLAN (1), base GVRP VLAN (4093), and Single STP VLAN (4094). • DYNAMIC – The VLAN was learned through GVRP.
VLAN-INDEX	A number used as an index into the internal database.

To display detailed information for a specific VLAN, enter a command such as the following:

```
FastIron#show gvrp vlan 1001

VLAN-ID: 1001, VLAN-INDEX: 7, STATIC: NO, DEFAULT: NO, BASE-VLAN: NO
Timer to Delete Entry Running: NO
Legend: [S=Slot]

Forbidden Members: None

Fixed Members: None

Normal(Dynamic) Members: (S2)  1
```

This display shows the following information.

Table 16.4: CLI Display of Summary VLAN Information for GVRP

This Field...	Displays...
VLAN-ID	The VLAN ID.
VLAN-INDEX	A number used as an index into the internal database.

Table 16.4: CLI Display of Summary VLAN Information for GVRP (Continued)

This Field...	Displays...
STATIC	Whether the VLAN is a statically configured VLAN.
DEFAULT	Whether this is the default VLAN.
BASE-VLAN	Whether this is the base VLAN for GVRP.
Timer to Delete Entry Running	Whether all ports have left the VLAN and the timer to delete the VLAN itself is running. The timer is described in the note for the Leave timer in “Changing the GVRP Timers” on page 16-7.
Legend	The meanings of the letter codes used in other parts of the display.
Forbidden Members	The ports that cannot become members of a VLAN advertised or leaned by GVRP.
Fixed Members	The ports that are statically configured members of the VLAN. GVRP cannot remove these ports.
Normal(Dynamic) Members	The ports that were added by GVRP. These ports also can be removed by GVRP.
MODE	The type of VLAN, which can be one of the following: • STATIC – The VLAN is statically configured and cannot be removed by GVRP. This includes VLANs you have configured as well as the default VLAN (1), base GVRP VLAN (4093), and Single STP VLAN (4094). • DYNAMIC – The VLAN was learned through GVRP.

To display detailed information for all VLANs, enter the **show gvrp vlan all** command.

Displaying GVRP Statistics

To display GVRP statistics for a port, enter a command such as the following:

```
FastIron#show gvrp statistics ethernet 2/1
PORT 2/1 Statistics:
 Leave All Received                     : 147
 Join Empty Received                    : 4193
 Join In Received                       : 599
 Leave Empty Received                   : 0
 Leave In Received                      : 0
 Empty Received                         : 588
 Leave All Transmitted                  : 157
 Join Empty Transmitted                 : 1794
 Join In Transmitted                    : 598
 Leave Empty Transmitted                : 0
 Leave In Transmitted                   : 0
 Empty Transmitted                      : 1248
 Invalid Messages/Attributes Skipped    : 0
 Failed Registrations                   : 0
```

Syntax: show gvrp statistics all | ethernet [<slotnum>/] <port-num>

This display shows the following information for the port.

Table 16.5: CLI Display of GVRP Statistics

This Field...	Displays...
Leave All Received	The number of Leaveall messages received.
Join Empty Received	The number of Join Empty messages received.
Join In Received	The number of Join In messages received.
Leave Empty Received	The number of Leave Empty messages received.
Leave In Received	The number of Leave In messages received.
Empty Received	The number of Empty messages received.
Leave All Transmitted	The number of Leaveall messages sent.
Join Empty Transmitted	The number of Join Empty messages sent.
Join In Transmitted	The number of Join In messages sent.
Leave Empty Transmitted	The number of Leave Empty messages sent.
Leave In Transmitted	The number of Leave In messages sent.
Empty Transmitted	The number of Empty messages sent.
Invalid Messages/Attributes Skipped	The number of invalid messages or attributes received or skipped. This can occur in the following cases: • The incoming GVRP PDU has an incorrect length. • "End of PDU" was reached before the complete attribute could be parsed. • The Attribute Type of the attribute that was being parsed was not the GVRP VID Attribute Type (0x01). • The attribute that was being parsed had an invalid attribute length. • The attribute that was being parsed had an invalid GARP event. • The attribute that was being parsed had an invalid VLAN ID. The valid range is 1 – 4095.
Failed Registrations	The number of failed registrations that have occurred. A failed registration can occur for the following reasons: • Join requests were received on a port that was blocked from learning dynamic VLANs (GVRP Blocking state). • An entry for a new GVRP VLAN could not be created in the GVRP database.

To display GVRP statistics for all ports, enter the **show gvrp statistics all** command.

Displaying CPU Utilization Statistics

You can display CPU utilization statistics for GVRP.

To display CPU utilization statistics for GVRP for the previous one-second, one-minute, five-minute, and fifteen-minute intervals, enter the following command at any level of the CLI:

```
FastIron#show process cpu
Process Name   5Sec(%)   1Min(%)   5Min(%)   15Min(%)   Runtime(ms)
ARP              0.01      0.03      0.09      0.22              9
BGP              0.00      0.00      0.00      0.00              0
GVRP             0.00      0.03      0.04      0.07              4
ICMP             0.00      0.00      0.00      0.00              0
IP               0.00      0.00      0.00      0.00              0
OSPF             0.00      0.00      0.00      0.00              0
RIP              0.00      0.00      0.00      0.00              0
STP              0.00      0.00      0.00      0.00              0
VRRP             0.00      0.00      0.00      0.00              0
```

If the software has been running less than 15 minutes (the maximum interval for utilization statistics), the command indicates how long the software has been running. Here is an example:

```
FastIron#show process cpu
The system has only been up for 6 seconds.
Process Name   5Sec(%)   1Min(%)   5Min(%)   15Min(%)   Runtime(ms)
ARP              0.01      0.00      0.00      0.00              0
BGP              0.00      0.00      0.00      0.00              0
GVRP             0.00      0.00      0.00      0.00              0
ICMP             0.01      0.00      0.00      0.00              1
IP               0.00      0.00      0.00      0.00              0
OSPF             0.00      0.00      0.00      0.00              0
RIP              0.00      0.00      0.00      0.00              0
STP              0.00      0.00      0.00      0.00              0
VRRP             0.00      0.00      0.00      0.00              0
```

To display utilization statistics for a specific number of seconds, enter a command such as the following:

```
FastIron#show process cpu 2
Statistics for last 1 sec and 80 ms
Process Name   Sec(%)   Time(ms)
ARP              0.00        0
BGP              0.00        0
GVRP             0.01        1
ICMP             0.00        0
IP               0.00        0
OSPF             0.00        0
RIP              0.00        0
STP              0.01        1
VRRP             0.00        0
```

When you specify how many seconds' worth of statistics you want to display, the software selects the sample that most closely matches the number of seconds you specified. In this example, statistics are requested for the previous two seconds. The closest sample available is actually for the previous 1 second plus 80 milliseconds.

Syntax: show process cpu [<num>]

The <num> parameter specifies the number of seconds and can be from 1 – 900. If you use this parameter, the command lists the usage statistics only for the specified number of seconds. If you do not use this parameter, the command lists the usage statistics for the previous one-second, one-minute, five-minute, and fifteen-minute intervals.

Displaying GVRP Diagnostic Information

To display diagnostic information, enter the following command:

```
FastIron#debug gvrp packets
         GVRP:  Packets debugging is on
GVRP: 0x2095ced4:  01 80 c2 00 00 21 00 e0 52 ab 87 40 00 3a 42 42
GVRP: 0x2095cee4:  03 00 01 01 02 00 04 05 00 02 04 05 00 07 04 05
GVRP: 0x2095cef4:  00 09 04 05 00 0b 04 02 03 e9 04 01 03 eb 04 01
GVRP: 0x2095cf04:  03 ec 04 01 03 ef 04 01 03 f1 04 01 05 dd 04 01
GVRP: 0x2095cf14:  09 cb 04 01 0f a1 00 00
GVRP: Port 2/1 RCV
GVRP: 0x2095ced4:  01 80 c2 00 00 21 00 e0 52 ab 87 40 00 28 42 42
GVRP: 0x2095cee4:  03 00 01 01 04 02 03 e9 04 01 03 eb 04 01 03 ec
GVRP: 0x2095cef4:  04 01 03 ef 04 01 03 f1 04 01 05 dd 04 01 09 cb
GVRP: 0x2095cf04:  04 01 0f a1 00 00
GVRP: Port 2/1 TX
GVRP: 0x207651b8:  01 80 c2 00 00 21 00 04 80 2c 0e 20 00 3a 42 42
GVRP: 0x207651c8:  03 00 01 01 02 00 04 05 03 e9 04 05 03 eb 04 05
GVRP: 0x207651d8:  03 ec 04 05 03 ef 04 05 03 f1 04 05 05 dd 04 05
GVRP: 0x207651e8:  09 cb 04 05 0f a1 04 02 00 02 04 01 00 07 04 01
GVRP: 0x207651f8:  00 09 04 01 00 0b 00 00
GVRP: Port 2/1 TX
GVRP: 0x207651b8:  01 80 c2 00 00 21 00 04 80 2c 0e 20 00 18 42 42
GVRP: 0x207651c8:  03 00 01 01 04 02 00 02 04 01 00 07 04 01 00 09
GVRP: 0x207651d8:  04 01 00 0b 00 00
```

Syntax: debug gvrp packets

Clearing GVRP Statistics

To clear the GVRP statistics counters, enter a command such as the following:

```
FastIron#clear gvrp statistics all
```

This command clears the counters for all ports. To clear the counters for a specific port only, enter a command such as the following:

```
FastIron#clear gvrp statistics ethernet 2/1
```

Syntax: clear gvrp statistics all | ethernet [<slotnum>/]<portnum>

CLI Examples

The following sections show the CLI commands for implementing the applications of GVRP described in “Application Examples” on page 16-1.

NOTE: Although some of the devices in these configuration examples do not have statically configured VLANs, this is not a requirement. You always can have statically configured VLANs on a device that is running GVRP.

Dynamic Core and Fixed Edge

In this configuration, the edge devices advertise their statically configured VLANs to the core device. The core device does not have any statically configured VLANs but learns the VLANs from the edge devices.

Enter the following commands on the core device:

```
FastIron> enable
FastIron#configure terminal
FastIron(config)#gvrp-enable
FastIron(config-gvrp)#enable all
```

These commands globally enable GVRP support and enable the protocol on all ports.

Enter the following commands on edge device A:

```
FastIron> enable
FastIron#configure terminal
FastIron(config)#vlan 20
FastIron(config-vlan-20)#untag ethernet 2/1
FastIron(config-vlan-20)#tag ethernet 4/24
FastIron(config-vlan-20)#vlan 40
FastIron(config-vlan-40)#untag ethernet 2/1
FastIron(config-vlan-40)#tag ethernet 4/24
FastIron(config-vlan-40)#exit
FastIron(config)#gvrp-enable
FastIron(config-gvrp)#enable ethernet 4/24
FastIron(config-gvrp)#block-learning ethernet 4/24
```

These commands statically configure two port-based VLANs, enable GVRP on port 4/24, and block GVRP learning on the port. The device will advertise the VLANs but will not learn VLANs from other devices.

Enter the following commands on edge device B:

```
FastIron> enable
FastIron#configure terminal
FastIron(config)#vlan 20
FastIron(config-vlan-20)#untag ethernet 2/24
FastIron(config-vlan-20)#tag ethernet 4/1
FastIron(config-vlan-20)#vlan 30
FastIron(config-vlan-30)#untag ethernet 4/24
FastIron(config-vlan-30)#tag ethernet 4/1
FastIron(config-vlan-30)#exit
FastIron(config)#gvrp-enable
FastIron(config-gvrp)#enable ethernet 4/1
FastIron(config-gvrp)#block-learning ethernet 4/1
```

Enter the following commands on edge device C:

```
FastIron> enable
FastIron#configure terminal
FastIron(config)#vlan 30
FastIron(config-vlan-30)#untag ethernet 2/24
FastIron(config-vlan-30)#tag ethernet 4/1
FastIron(config-vlan-20)#vlan 40
FastIron(config-vlan-40)#untag ethernet 4/24
FastIron(config-vlan-40)#tag ethernet 4/1
FastIron(config-vlan-40)#exit
FastIron(config)#gvrp-enable
FastIron(config-gvrp)#enable ethernet 4/1
FastIron(config-gvrp)#block-learning ethernet 4/1
```

Dynamic Core and Dynamic Edge

In this configuration, the core and edge devices have no statically configured VLANs and are enabled to learn and advertise VLANs. The edge and core devices learn the VLANs configured on the devices in the edge clouds. To enable GVRP on all the ports, enter the following command on each edge device ***and*** on the core device.

```
FastIron> enable
FastIron#configure terminal
FastIron(config)#gvrp-enable
FastIron(config-gvrp)#enable all
```

Fixed Core and Dynamic Edge

In this configuration, GVRP learning is enabled on the edge devices. The VLANs on the core device are statically configured, and the core device is enabled to advertise its VLANs but not to learn VLANs. The edge devices learn the VLANs from the core.

Enter the following commands on the core device:

```
FastIron> enable
FastIron#configure terminal
FastIron(config)#vlan 20
FastIron(config-vlan-20)#tag ethernet 1/24
FastIron(config-vlan-20)#tag ethernet 6/24
FastIron(config-vlan-20)#vlan 30
FastIron(config-vlan-30)#tag ethernet 6/24
FastIron(config-vlan-30)#tag ethernet 8/17
FastIron(config-vlan-30)#vlan 40
FastIron(config-vlan-40)#tag ethernet 1/5
FastIron(config-vlan-40)#tag ethernet 8/17
FastIron(config-vlan-40)#vlan 50
FastIron(config-vlan-50)#untag ethernet 6/1
FastIron(config-vlan-50)#tag ethernet 1/11
FastIron(config-vlan-50)#exit
FastIron(config)#gvrp-enable
FastIron(config-gvrp)#enable ethernet 1/24 ethernet 6/24 ethernet 8/17
FastIron(config-gvrp)#block-learning ethernet 1/24 ethernet 6/24 ethernet 8/17
```

These VLAN commands configure VLANs 20, 30, 40, and 50. The GVRP commands enable the protocol on the ports that are connected to the edge devices, and disable VLAN learning on those ports. All the VLANs are advertised by GVRP.

Enter the following commands on edge devices A, B, and C:

```
FastIron> enable
FastIron#configure terminal
FastIron(config)#gvrp-enable
FastIron(config-gvrp)#enable all
FastIron(config-gvrp)#block-applicant all
```

Fixed Core and Fixed Edge

The VLANs are statically configured on the core and edge devices. On each edge device, VLAN advertising is enabled but learning is disabled. GVRP is not configured on the core device. This configuration enables the devices in the edge clouds to learn the VLANs configured on the edge devices.

This configuration does not use any GVRP configuration on the core device.

The configuration on the edge device is the same as in “Dynamic Core and Fixed Edge” on page 16-17.

Chapter 17

Configuring Rule-Based IP Access Control Lists (ACLs)

This chapter describes how Access Control Lists (ACLs) are implemented and configured in the FastIron devices.

NOTE: For information about IPv6 ACLs, see the chapter "Configuring IPv6 Access Control Lists (ACLs)" on page 18-1.

ACL Overview

Foundry's FastIron devices support ***rule-based ACLs*** (sometimes called hardware-based ACLs), where the decisions to permit or deny packets are processed in hardware and all permitted packets are switched or routed in hardware. In addition, Foundry's FastIron devices support inbound ACLs only. Outbound ACL are not supported.

NOTE: Foundry's FastIron devices support hardware-based ACLs only. These devices do not support flow-based ACLs. In contrast, FES devices support flow-based ACLs only.

Rule-based ACLs program the ACL entries you assign to an interface into Content Addressable Memory (CAM) space allocated for the ports. The ACLs are programmed into hardware at startup (or as new ACLs are entered and bound to ports). Devices that use rule-based ACLs program the ACLs into the CAM entries and use these entries to permit or deny packets in the hardware, without sending the packets to the CPU for processing.

Rule-based ACLs are supported on the following interface types:

- Gigabit Ethernet ports
- 10-Gigabit Ethernet ports
- Trunk groups
- Virtual routing interfaces

Types of IP ACLs

You can configure the following types of IP ACLs:

- ***Standard*** – Permits or denies packets based on source IP address. Valid standard ACL IDs are 1 – 99 or a character string.
- ***Extended*** – Permits or denies packets based on source and destination IP address and also based on IP protocol information. Valid extended ACL IDs are a number from 100 – 199 or a character string.

ACL IDs and Entries

ACLs consist of ACL IDs and ACL entries:

- ***ACL ID*** – An ACL ID is a number from 1 – 99 (for a standard ACL) or 100 – 199 (for an extended ACL) or a character string. The ACL ID identifies a collection of individual ACL entries. When you apply ACL entries to an interface, you do so by applying the ACL ID that contains the ACL entries to the interface, instead of applying the individual entries to the interface. This makes applying large groups of access filters (ACL entries) to interfaces simple. See also “Numbered and Named ACLs” on page 17-2.

  NOTE: This is different from IP access policies. If you use IP access policies, you apply the individual policies to interfaces.

- ***ACL entry*** – Also called an ***ACL rule***, a filter command associated with an ACL ID. The maximum number of ACL rules you can configure is a system-wide parameter and depends on the device you are configuring. You can configure up to the maximum number of entries in any combination in different ACLs. The total number of entries in all ACLs cannot exceed the system maximum.
  - The maximum number of ACL entries supported system-wide in the Layer 2 switch code is 1015 in releases 02.5.00 and earlier, and 8192 starting in release 03.0.00.
  - The maximum number of ACL entries supported system-wide in the Layer 3 router code is 4096 in releases 2.5.00 and earlier, and 8192 starting in release 03.0.00. The default value is 2048

  NOTE: For FastIron GS and LS devices, the maximum number of ACLs that can be configured is 1021.

  - One-Gigabit ports support up to 1016 ACL rules per port region (12 ports per GbE port region). Each ACL group must contain one entry for the implicit *deny all IP traffic* clause. Also, each ACL group uses a multiple of 8 ACL entries. For example, if all ACL groups contain 5 ACL entries, you could add 127ACL groups (1016/8) in that port region. If all your ACL groups contain 8 ACL entries, you could add 63 ACL groups, since you must account for the implicit deny entry.
  - 10-Gigabit ports support up to 1024 ACL rules per port region (1 port per 10-GbE port region).

You configure ACLs on a global basis, then apply them to the incoming traffic on specific ports. The software applies the entries within an ACL in the order they appear in the ACL’s configuration. As soon as a match is found, the software takes the action specified in the ACL entry (permit or deny the packet) and stops further comparison for that packet.

Numbered and Named ACLs

When you configure an ACL, you can refer to the ACL by a numeric ID or by an alphanumeric name. The commands to configure numbered ACLs are different from the commands for named ACLs.

- ***Numbered ACL*** – If you refer to the ACL by a numeric ID, you can use 1 – 99 for a standard ACL or 100 – 199 for an extended ACL.
- ***Named ACL*** – If you refer to the ACL by a name, you specify whether the ACL is a standard ACL or an extended ACL, then specify the name.

You can configure up to 99 standard numbered IP ACLs and 99 extended numbered IP ACLs. You also can configure up to 99 standard named ACLs and 99 extended named ACLs by number. Regardless of how many ACLs you have, the device can have a maximum of 1024 ACL entries, associated with the ACLs in any combination.

Default ACL Action

The default action when no ACLs are configured on a device is to permit all traffic. However, once you configure an ACL and apply it to a port, the default action for that port is to deny all traffic that is not explicitly permitted on the port.

- If you want to tightly control access, configure ACLs consisting of permit entries for the access you want to permit. The ACLs implicitly deny all other access.

- If you want to secure access in environments with many users, you might want to configure ACLs that consist of explicit deny entries, then add an entry to permit all access to the end of each ACL. The software permits packets that are not denied by the deny entries.

How Hardware-Based ACLs Work

When you bind an ACL to inbound traffic on an interface, the device programs the Layer 4 CAM with the ACL. Permit and deny rules are programmed. Most ACL rules require one Layer 4 CAM entry. However, ACL rules that match on more than one TCP or UDP application port may require several CAM entries. The Layer 4 CAM entries for ACLs do not age out. They remain in the CAM until you remove the ACL.

- If a packet received on the interface matches an ACL rule in the Layer 4 CAM, the device permits or denies the packet according to the ACL.
- If a packet does not match an ACL rule, the packet is dropped, since the default action on an interface that has ACLs is to deny the packet.

How Fragmented Packets are Processed

The descriptions above apply to non-fragmented packets. The default processing of fragments by hardware-based ACLs is as follows:

- The first fragment of a packet is permitted or denied using the ACLs. The first fragment is handled the same way as non-fragmented packets, since the first fragment contains the Layer 4 source and destination application port numbers. The device uses the Layer 4 CAM entry if one is programmed, or applies the interface's ACL entries to the packet and permits or denies the packet according to the first matching ACL.
- For other fragments of the same packet, they are subject to a rule only if there is no Layer 4 information in the rule or in any preceding rules.

The fragments are forwarded even if the first fragment, which contains the Layer 4 information, was denied. Generally, denying the first fragment of a packet is sufficient, since a transaction cannot be completed without the entire packet.

For tighter control, you can configure the port to drop all packet fragments. See “Enabling Strict Control of ACL Filtering of Fragmented Packets” on page 17-22.

Hardware Aging of Layer 4 CAM Entries

Rule-based ACLs use Layer 4 CAM entries. The device permanently programs rule-based ACLs into the CAM. The entries never age out.

Configuration Considerations

- Foundry’s FastIron devices support inbound ACLs. Outbound ACL are not supported.
- Hardware-based ACLs are supported on:
  - Gigabit Ethernet ports
  - 10-Gigabit Ethernet ports
  - Trunk groups
  - Virtual routing interfaces
- ACLs on the FastIron X Series devices apply to all traffic, including management traffic.
- ACL logging is supported for packets that are sent to the CPU for processing (denied packets). ACL logging is not supported for packets that are processed in hardware (permitted packets).
- Hardware-based ACLs support only one ACL per port. The ACL of course can contain multiple entries (rules). For example, hardware-based ACLs do not support ACLs 101 and 102 on port 1, but hardware-based ACLs do support ACL 101 containing multiple entries.

- The number of ACL rules supported is as follows:
  - FastIron X Series devices – 1-Gigabit ports support up to 1016 ACL rules and 10-Gigabit ports support up to 1024 ACL rules.
  - FastIron GS devices – 1-Gigabit and 10-Gigabit ports support up to 1021 ACL rules.
- ACLs on the FSX are affected by port regions. Multiple ACL groups share 1016 ACL rules per port region. Each ACL group must contain one entry for the implicit *deny all IP traffic* clause. Also, each ACL group uses a multiple of 8 ACL entries. For example, if all ACL groups contain 5 ACL entries, you could add 127ACL groups (1016/8) in that port region. If all your ACL groups contain 8 ACL entries, you could add 63 ACL groups, since you must account for the implicit deny entry.
- By default, the first fragment of a fragmented packet received by the Foundry device is permitted or denied using the ACLs, but subsequent fragments of the same packet are forwarded in hardware. Generally, denying the first fragment of a packet is sufficient, since a transaction cannot be completed without the entire packet.
- Starting in software release FSX 04.0.01, ACLs are supported on member ports of a VLAN on which DHCP snooping and Dynamic ARP Inspection (DAI) are enabled. Also new in release FSX 04.0.01, IP source guard and ACLs are supported together on the same port, as long as both features are configured at the port-level or per-port-per-VLAN level. Foundry ports do not support IP source guard and ACLs on the same port if one is configured at the port-level and the other is configured at the per-pert-per-VLAN level.
- The following ACL features and options are not supported on the FastIron devices:
  - Applying an ACL on a device that has Super Aggregated VLANs (SAVs) enabled.
  - ACL logging – ACL logging is supported for packets that are sent to the CPU for processing (denied packets). ACL logging is not supported for packets that are processed in hardware (permitted packets).
  - Flow-based ACLs
  - ACL statistics
  - Layer 2 ACLs

NOTE: You can apply an ACL to a port that has TCP SYN protection and/or ICMP smurf protection enabled.

Configuring Standard Numbered ACLs

This section describes how to configure standard numbered ACLs with numeric IDs and provides configuration examples.

Standard ACLs permit or deny packets based on source IP address. You can configure up to 99 standard numbered ACLs. There is no limit to the number of ACL entries an ACL can contain except for the system-wide limitation. For the number of ACL entries supported on a device, see “ACL IDs and Entries” on page 17-2.

Standard Numbered ACL Syntax

Syntax: [no] access-list <acl-num> deny | permit <source-ip> | <hostname> <wildcard> [log]

or

Syntax: [no] access-list <acl-num> deny | permit <source-ip>/<mask-bits> | <hostname> [log]

Syntax: [no] access-list <acl-num> deny | permit host <source-ip> | <hostname> [log]

Syntax: [no] access-list <acl-num> deny | permit any [log]

Syntax: [no] ip access-group <acl-num> in

The <acl-num> parameter is the access list number from 1 – 99.

The **deny | permit** parameter indicates whether packets that match a policy in the access list are denied (dropped) or permitted (forwarded).

The <source-ip> parameter specifies the source IP address. Alternatively, you can specify the host name.

NOTE: To specify the host name instead of the IP address, the host name must be configured using the Foundry device's DNS resolver. To configure the DNS resolver name, use the **ip dns server-address…** command at the global CONFIG level of the CLI.

The <wildcard> parameter specifies the mask value to compare against the host address specified by the <source-ip> parameter. The <wildcard> is a four-part value in dotted-decimal notation (IP address format) consisting of ones and zeros. Zeros in the mask mean the packet's source address must match the <source-ip>. Ones mean any value matches. For example, the <source-ip> and <wildcard> values 209.157.22.26 0.0.0.255 mean that all hosts in the Class C subnet 209.157.22.x match the policy.

If you prefer to specify the wildcard (mask value) in CIDR format, you can enter a forward slash after the IP address, then enter the number of significant bits in the mask. For example, you can enter the CIDR equivalent of "209.157.22.26 0.0.0.255" as "209.157.22.26/24". The CLI automatically converts the CIDR number into the appropriate ACL mask (where zeros instead of ones are the significant bits) and changes the non-significant portion of the IP address into ones. For example, if you specify 209.157.22.26/24 or 209.157.22.26 0.0.0.255, then save the changes to the startup-config file, the value appears as 209.157.22.0/24 (if you have enabled display of subnet lengths) or 209.157.22.0 0.0.0.255 in the startup-config file.

If you enable the software to display IP subnet masks in CIDR format, the mask is saved in the file in "/<mask-bits>" format. To enable the software to display the CIDR masks, enter the **ip show-subnet-length** command at the global CONFIG level of the CLI. You can use the CIDR format to configure the ACL entry regardless of whether the software is configured to display the masks in CIDR format.

NOTE: If you use the CIDR format, the ACL entries appear in this format in the running-config and startup-config files, but are shown with subnet mask in the display produced by the **show ip access-list** command.

The **host** <source-ip> | <hostname> parameter lets you specify a host IP address or name. When you use this parameter, you do not need to specify the mask. A mask of all zeros (0.0.0.0) is implied.

The **any** parameter configures the policy to match on all host addresses.

The **log** argument configures the device to generate Syslog entries and SNMP traps for packets that are denied by the access policy.

NOTE: You can enable logging on ACLs and filters that support logging even when the ACLs and filters are already in use. To do so, re-enter the ACL or filter command and add the **log** parameter to the end of the ACL or filter. The software replaces the ACL or filter command with the new one. The new ACL or filter, with logging enabled, takes effect immediately.

The **in** parameter applies the ACL to incoming traffic on the interface to which you apply the ACL. You can apply the ACL to an Ethernet port or virtual interface.

NOTE: If the ACL is for a virtual routing interface, you also can specify a subset of ports within the VLAN containing that interface when assigning an ACL to the interface.

Configuration Example for Standard Numbered ACLs

To configure a standard ACL and apply it to incoming traffic on port 1/1, enter the following commands.

```
FastIron(config)#access-list 1 deny host 209.157.22.26 log
FastIron(config)#access-list 1 deny 209.157.29.12 log
FastIron(config)#access-list 1 deny host IPHost1 log
FastIron(config)#access-list 1 permit any
FastIron(config)#int eth 1/1
FastIron(config-if-1/1)#ip access-group 1 in
FastIron(config)#write memory
```

The commands in this example configure an ACL to deny packets from three source IP addresses from being received on port 1/1. The last ACL entry in this ACL permits all packets that are not explicitly denied by the first three ACL entries.

Configuring Standard Named ACLs

This section describes how to configure standard named ACLs with alphanumeric IDs. This section also provides configuration examples.

Standard ACLs permit or deny packets based on source IP address. You can configure up to 99 standard named ACLs. There is no limit to the number of ACL entries an ACL can contain except for the system-wide limitation. For the number of ACL entries supported on a device, see "ACL IDs and Entries" on page 17-2.

The commands for configuring named ACL entries are different from the commands for configuring numbered ACL entries. The command to configure a numbered ACL is **access-list**. The command for configuring a named ACL is **ip access-list**. In addition, when you configure a numbered ACL entry, you specify all the command parameters on the same command. When you configure a named ACL, you specify the ACL type (standard or extended) and the ACL name with one command, which places you in the configuration level for that ACL. Once you enter the configuration level for the ACL, the command syntax is the same as the syntax for numbered ACLs.

Standard Named ACL Syntax

Syntax: [no] ip access-list standard <acl-name> | <acl-num>

Syntax: deny | permit <source-ip> | <hostname> <wildcard> [log]

or

Syntax: deny | permit <source-ip>/<mask-bits> | <hostname> [log]

Syntax: deny | permit host <source-ip> | <hostname> [log]

Syntax: deny | permit any [log]

Syntax: [no] ip access-group <acl-name> in

The <acl-name> parameter is the access list name. You can specify a string of up to 256 alphanumeric characters. You can use blanks in the ACL name if you enclose the name in quotation marks (for example, "ACL for Net1").

The <acl-num> parameter allows you to specify an ACL number if you prefer. If you specify a number, you can specify from 1 – 99 for standard ACLs.

NOTE: For convenience, the software allows you to configure numbered ACLs using the syntax for named ACLs. The software also still supports the older syntax for numbered ACLs. Although the software allows both methods for configuring numbered ACLs, numbered ACLs are always formatted in the startup-config and running-config files in using the older syntax, as follows.

```
access-list 1 deny host 209.157.22.26 log
access-list 1 deny 209.157.22.0 0.0.0.255 log
access-list 1 permit any
access-list 101 deny tcp any any eq http log
```

The **deny | permit** parameter indicates whether packets that match a policy in the access list are denied (dropped) or permitted (forwarded).

The <source-ip> parameter specifies the source IP address. Alternatively, you can specify the host name.

NOTE: To specify the host name instead of the IP address, the host name must be configured using the Foundry device's DNS resolver. To configure the DNS resolver name, use the **ip dns server-address…** command at the global CONFIG level of the CLI.

The <wildcard> parameter specifies the mask value to compare against the host address specified by the <source-ip> parameter. The <wildcard> is a four-part value in dotted-decimal notation (IP address format) consisting of ones and zeros. Zeros in the mask mean the packet's source address must match the <source-ip>. Ones mean any value matches. For example, the <source-ip> and <wildcard> values 209.157.22.26 0.0.0.255 mean that all hosts in the Class C subnet 209.157.22.x match the policy.

If you prefer to specify the wildcard (mask value) in CIDR format, you can enter a forward slash after the IP address, then enter the number of significant bits in the mask. For example, you can enter the CIDR equivalent of "209.157.22.26 0.0.0.255" as "209.157.22.26/24". The CLI automatically converts the CIDR number into the appropriate ACL mask (where zeros instead of ones are the significant bits) and changes the non-significant portion of the IP address into ones. For example, if you specify 209.157.22.26/24 or 209.157.22.26 0.0.0.255, then save the changes to the startup-config file, the value appears as 209.157.22.0/24 (if you have enabled display of subnet lengths) or 209.157.22.0 0.0.0.255 in the startup-config file.

If you enable the software to display IP subnet masks in CIDR format, the mask is saved in the file in "/<mask-bits>" format. To enable the software to display the CIDR masks, enter the **ip show-subnet-length** command at the global CONFIG level of the CLI. You can use the CIDR format to configure the ACL entry regardless of whether the software is configured to display the masks in CIDR format.

NOTE: If you use the CIDR format, the ACL entries appear in this format in the running-config and startup-config files, but are shown with subnet mask in the display produced by the **show ip access-list** command.

The **host** <source-ip> | <hostname> parameter lets you specify a host IP address or name. When you use this parameter, you do not need to specify the mask. A mask of all zeros (0.0.0.0) is implied.

The **any** parameter configures the policy to match on all host addresses.

The **log** argument configures the device to generate Syslog entries and SNMP traps for packets that are denied by the access policy.

NOTE: You can enable logging on ACLs and filters that support logging even when the ACLs and filters are already in use. To do so, re-enter the ACL or filter command and add the **log** parameter to the end of the ACL or filter. The software replaces the ACL or filter command with the new one. The new ACL or filter, with logging enabled, takes effect immediately.

The **in** parameter applies the ACL to incoming traffic on the interface to which you apply the ACL. You can apply the ACL to an Ethernet port or virtual interface.

NOTE: If the ACL is bound to a virtual routing interface, you also can specify a subset of ports within the VLAN containing that interface when assigning an ACL to the interface.

Configuration Example for Standard Named ACLs

To configure a standard named ACL, enter commands such as the following.

```
FastIron(config)#ip access-list standard Net1
FastIron(config-std-nacl)#deny host 209.157.22.26 log
FastIron(config-std-nacl)#deny 209.157.29.12 log
FastIron(config-std-nacl)#deny host IPHost1 log
FastIron(config-std-nacl)#permit any
FastIron(config-std-nacl)#exit
FastIron(config)#int eth 1/1
FastIron(config-if-e1000-1/1)#ip access-group Net1 in
```

The commands in this example configure a standard ACL named “Net1”. The entries in this ACL deny packets from three source IP addresses from being forwarded on port 1. Since the implicit action for an ACL is “deny”, the last ACL entry in this ACL permits all packets that are not explicitly denied by the first three ACL entries. For an example of how to configure the same entries in a numbered ACL, see “Configuring Standard Numbered ACLs” on page 17-4.

Notice that the command prompt changes after you enter the ACL type and name. The “std” in the command prompt indicates that you are configuring entries for a standard ACL. For an extended ACL, this part of the command prompt is “ext“. The “nacl” indicates that you are configuring a named ACL.

Configuring Extended Numbered ACLs

This section describes how to configure extended numbered ACLs.

Extended ACLs let you permit or deny packets based on the following information:

- IP protocol
- Source IP address or host name
- Destination IP address or host name
- Source TCP or UDP port (if the IP protocol is TCP or UDP)
- Destination TCP or UDP port (if the IP protocol is TCP or UDP)

The IP protocol can be one of the following well-known names or any IP protocol number from 0 – 255:

- Internet Control Message Protocol (ICMP)
- Internet Group Management Protocol (IGMP)
- Internet Gateway Routing Protocol (IGRP)
- Internet Protocol (IP)
- Open Shortest Path First (OSPF)
- Transmission Control Protocol (TCP)
- User Datagram Protocol (UDP)

For TCP and UDP, you also can specify a comparison operator and port name or number. For example, you can configure a policy to block web access to a specific website by denying all TCP port 80 (HTTP) packets from a specified source IP address to the website’s IP address.

Extended Numbered ACL Syntax

[no] access-list <acl-num> deny | permit <ip-protocol> <source-ip> | <hostname> <wildcard> [<operator> <source-tcp/udp-port>] <destination-ip> | <hostname> [<icmp-num> | <icmp-type>] <wildcard> [<tcp/udp comparison operator> <destination-tcp/udp-port>] [dscp-cos-mapping] [dscp-marking <0-63> [802.1p-priority-

marking <0 –7>... | dscp-cos-mapping]] [dscp-matching <0-63>] [log] [precedence <name> | <0 – 7>] [tos <0 – 63> | <name>] [traffic policy <name>]

[no] access-list <acl-num> deny | permit host <ip-protocol> any any

Syntax: [no] ip access-group <acl-num> in

The <acl-num> parameter is the extended access list number. Specify a number from 100 – 199.

The **deny | permit** parameter indicates whether packets that match the policy are dropped or forwarded.

The <ip-protocol> parameter indicates the type of IP packet you are filtering. You can specify a well-known name for any protocol whose number is less than 255. For other protocols, you must enter the number. Enter "?" instead of a protocol to list the well-known names recognized by the CLI.

The <source-ip> | <hostname> parameter specifies the source IP host for the policy. If you want the policy to match on all source addresses, enter **any**.

The <wildcard> parameter specifies the portion of the source IP host address to match against. The <wildcard> is a four-part value in dotted-decimal notation (IP address format) consisting of ones and zeros. Zeros in the mask mean the packet's source address must match the <source-ip>. Ones mean any value matches. For example, the <source-ip> and <wildcard> values 209.157.22.26 0.0.0.255 mean that all hosts in the Class C subnet 209.157.22.x match the policy.

If you prefer to specify the wildcard (mask value) in Classless Interdomain Routing (CIDR) format, you can enter a forward slash after the IP address, then enter the number of significant bits in the mask. For example, you can enter the CIDR equivalent of "209.157.22.26 0.0.0.255" as "209.157.22.26/24". The CLI automatically converts the CIDR number into the appropriate ACL mask (where zeros instead of ones are the significant bits) and changes the non-significant portion of the IP address into zeros. For example, if you specify 209.157.22.26/24 or 209.157.22.26 0.0.0.255, then save the changes to the startup-config file, the value appears as 209.157.22.0/24 (if you have enabled display of subnet lengths) or 209.157.22.0 0.0.0.255 in the startup-config file.

If you enable the software to display IP subnet masks in CIDR format, the mask is saved in the file in "/<mask-bits>" format. To enable the software to display the CIDR masks, enter the **ip show-subnet-length** command at the global CONFIG level of the CLI. You can use the CIDR format to configure the ACL entry regardless of whether the software is configured to display the masks in CIDR format.

NOTE: If you use the CIDR format, the ACL entries appear in this format in the running-config and startup-config files, but are shown with subnet mask in the display produced by the **show ip access-list** command.

The <destination-ip> | <hostname> parameter specifies the destination IP host for the policy. If you want the policy to match on all destination addresses, enter **any**.

The <icmp-type> | <icmp-num> parameter specifies the ICMP protocol type.

- This parameter applies only if you specified **icmp** as the <ip-protocol> value.
- If you use this parameter, the ACL entry is sent to the CPU for processing.
- If you do not specify a message type, the ACL applies to all types of ICMP messages.

The <icmp-num> parameter can be a value from 0 – 255.

The <icmp-type> parameter can have one of the following values, depending on the software version the device is running:

- any-icmp-type
- echo
- echo-reply
- information-request
- log
- mask-reply

- mask-request
- parameter-problem
- redirect
- source-quench
- time-exceeded
- timestamp-reply
- timestamp-request
- traffic policy
- unreachable
- <num>

The <tcp/udp comparison operator> parameter specifies a comparison operator for the TCP or UDP port number. This parameter applies only when you specify **tcp** or **udp** as the IP protocol. For example, if you are configuring an entry for HTTP, specify **tcp eq http**. You can enter one of the following operators:

- **eq** – The policy applies to the TCP or UDP port name or number you enter after **eq**.
- **established** – This operator applies only to TCP packets. If you use this operator, the policy applies to TCP packets that have the ACK (Acknowledgment) or RST (Reset) bits set on (set to "1") in the Control Bits field of the TCP packet header. Thus, the policy applies only to established TCP sessions, not to new sessions. See Section 3.1, "Header Format", in RFC 793 for information about this field.

  NOTE: This operator applies only to destination TCP ports, not source TCP ports.

- **gt** – The policy applies to TCP or UDP port numbers greater than the port number or the numeric equivalent of the port name you enter after **gt**.
- **lt** – The policy applies to TCP or UDP port numbers that are less than the port number or the numeric equivalent of the port name you enter after **lt**.
- **neq** – The policy applies to all TCP or UDP port numbers except the port number or port name you enter after **neq**.
- **range** – The policy applies to all TCP or UDP port numbers that are between the first TCP or UDP port name or number and the second one you enter following the **range** parameter. The range includes the port names or numbers you enter. For example, to apply the policy to all ports between and including 23 (Telnet) and 53 (DNS), enter the following: **range 23 53**. The first port number in the range must be lower than the last number in the range.

The <tcp/udp-port> parameter specifies the TCP or UDP port number or well-known name. You can specify a well-known name for any application port whose number is less than 1024. For other application ports, you must enter the number. Enter "?" instead of a port to list the well-known names recognized by the CLI.

The **in** parameter specifies that the ACL applies to incoming traffic on the interface to which you apply the ACL. You can apply the ACL to an Ethernet port or a virtual interface.

NOTE: If the ACL is for a virtual routing interface, you also can specify a subset of ports within the VLAN containing that interface when assigning an ACL to the interface. See "Configuring Standard Numbered ACLs" on page 17-4.

The **precedence** <name> | <num> parameter of the **ip access-list** command specifies the IP precedence. The precedence option for of an IP packet is set in a three-bit field following the four-bit header-length field of the packet's header. You can specify one of the following:

- **critical** or **5** – The ACL matches packets that have the critical precedence. If you specify the option number instead of the name, specify number 5.

- **flash** or **3** – The ACL matches packets that have the flash precedence. If you specify the option number instead of the name, specify number 3.
- **flash-override** or **4** – The ACL matches packets that have the flash override precedence. If you specify the option number instead of the name, specify number 4.
- **immediate** or **2** – The ACL matches packets that have the immediate precedence. If you specify the option number instead of the name, specify number 2.
- **internet** or **6** – The ACL matches packets that have the internetwork control precedence. If you specify the option number instead of the name, specify number 6.
- **network** or **7** – The ACL matches packets that have the network control precedence. If you specify the option number instead of the name, specify number 7.
- **priority** or **1** – The ACL matches packets that have the priority precedence. If you specify the option number instead of the name, specify number 1.
- **routine** or **0** – The ACL matches packets that have the routine precedence. If you specify the option number instead of the name, specify number 0.

The **tos** <name> | <num> parameter of the **ip access-list** command specifies the IP ToS. You can specify one of the following:

- **max-reliability** or **2** – The ACL matches packets that have the maximum reliability ToS. The decimal value for this option is 2.
- **max-throughput** or **4** – The ACL matches packets that have the maximum throughput ToS. The decimal value for this option is 4.
- **min-delay** or **8** – The ACL matches packets that have the minimum delay ToS. The decimal value for this option is 8.
- **min-monetary-cost** or **1** – The ACL matches packets that have the minimum monetary cost ToS. The decimal value for this option is 1.

  NOTE: This value is not supported on 10 Gigabit Ethernet modules.

- **normal** or **0** – The ACL matches packets that have the normal ToS. The decimal value for this option is 0.
- <num> – A number from 0 – 15 that is the sum of the numeric values of the options you want. The ToS field is a four-bit field following the Precedence field in the IP header. You can specify one or more of the following. To select more than one option, enter the decimal value that is equivalent to the sum of the numeric values of all the ToS options you want to select. For example, to select the **max-reliability** and **min-delay** options, enter number 10. To select all options, select 15.

The **dscp-cos-mapping** option maps the DSCP value in incoming packets to a hardware table that provides mapping of each of the 0 – 63 DSCP values, and distributes them among eight traffic classes (internal priorities) and eight 802.1p priorities.

NOTE: The **dscp-cos-mapping** option overrides port-based priority settings.

The **dscp-marking** option enables you to configure an ACL that marks matching packets with a specified DSCP value Enter a value from 0 – 63. See “Using an IP ACL to Mark DSCP Values (DSCP Marking)” on page 17-29.

The **dscp-matching** option matches on the packet’s DSCP value. Enter a value from 0 – 63. This option does not change the packet’s forwarding priority through the device or mark the packet. See “DSCP Matching” on page 17-30.

The **log** parameter enables SNMP traps and Syslog messages for packets denied by the ACL.

You can enable logging on ACLs and filters that support logging even when the ACLs and filters are already in use. To do so, re-enter the ACL or filter command and add the **log** parameter to the end of the ACL or filter. The software replaces the ACL or filter command with the new one. The new ACL or filter, with logging enabled, takes effect immediately.

The **traffic-policy** option enables the device to rate limit inbound traffic and to count the packets and bytes per packet to which ACL permit or deny clauses are applied. For configuration procedures and examples, see the chapter “Configuring Traffic Policies” on page 24-1.

Configuration Examples for Extended Numbered ACLs

To configure an extended access list that blocks all Telnet traffic received on port 1/1 from IP host 209.157.22.26, enter the following commands.

```
FastIron(config)#access-list 101 deny tcp host 209.157.22.26 any eq telnet log
FastIron(config)#access-list 101 permit ip any any
FastIron(config)#int eth 1/1
FastIron(config-if-e1000-1/1)#ip access-group 101 in
FastIron(config)#write memory
```

Here is another example of commands for configuring an extended ACL and applying it to an interface. These examples show many of the syntax choices. Notice that some of the entries are configured to generate log entries while other entries are not thus configured.

```
FastIron(config)#access-list 102 perm icmp 209.157.22.0/24 209.157.21.0/24
FastIron(config)#access-list 102 deny igmp host rkwong 209.157.21.0/24 log
FastIron(config)#access-list 102 deny igrp 209.157.21.0/24 host rkwong log
FastIron(config)#access-list 102 deny ip host 209.157.21.100 host 209.157.22.1 log
FastIron(config)#access-list 102 deny ospf any any log
FastIron(config)#access-list 102 permit ip any any
```

The first entry permits ICMP traffic from hosts in the 209.157.22.*x* network to hosts in the 209.157.21.*x* network.

The second entry denies IGMP traffic from the host device named “rkwong” to the 209.157.21.*x* network.

The third entry denies IGRP traffic from the 209.157.21.*x* network to the host device named “rkwong”.

The fourth entry denies all IP traffic from host 209.157.21.100to host 209.157.22.1 and generates Syslog entries for packets that are denied by this entry.

The fifth entry denies all OSPF traffic and generates Syslog entries for denied traffic.

The sixth entry permits all packets that are not explicitly denied by the other entries. Without this entry, the ACL would deny all incoming or outgoing IP traffic on the ports to which you assign the ACL.

The following commands apply ACL 102 to the incoming traffic on port 1/2 and to the incoming traffic on port 4/3.

```
FastIron(config)#int eth 1/2
FastIron(config-if-1/2)#ip access-group 102 in
FastIron(config-if-1/2)#exit
FastIron(config)#int eth 4/3
FastIron(config-if-4/3)#ip access-group 102 in
FastIron(config)#write memory
```

Here is another example of an extended ACL.

```
FastIron(config)#access-list 103 deny tcp 209.157.21.0/24 209.157.22.0/24
FastIron(config)#access-list 103 deny tcp 209.157.21.0/24 eq ftp 209.157.22.0/24
FastIron(config)#access-list 103 deny tcp 209.157.21.0/24 209.157.22.0/24 lt
telnet neq 5
FastIron(config)#access-list 103 deny udp any range 5 6 209.157.22.0/24 range 7 8
FastIron(config)#access-list 103 permit ip any any
```

The first entry in this ACL denies TCP traffic from the 209.157.21.*x* network to the 209.157.22.x network.

The second entry denies all FTP traffic from the 209.157.21.*x* network to the 209.157.22.x network.

The third entry denies TCP traffic from the 209.157.21.*x* network to the 209.157.22.*x* network, if the TCP port number of the traffic is less than the well-known TCP port number for Telnet (23), and if the TCP port is not equal to 5. Thus, TCP packets whose TCP port numbers are 5 or are greater than 23 are allowed.

The fourth entry denies UDP packets from any source to the 209.157.22.*x* network, if the UDP port number from the source network is 5 or 6 and the destination UDP port is 7 or 8.

The fifth entry permits all packets that are not explicitly denied by the other entries. Without this entry, the ACL would deny all incoming or outgoing IP traffic on the ports to which you assign the ACL.

The following commands apply ACL 103 to the incoming traffic on ports 2/1 and 2/2.

```
FastIron(config)#int eth 2/1
FastIron(config-if-2/1)#ip access-group 103 in
FastIron(config-if-2/1)#exit
FastIron(config)#int eth 0/2/2
FastIron(config-if-2/2)#ip access-group 103 in
FastIron(config)#write memory
```

Configuring Extended Named ACLs

The commands for configuring named ACL entries are different from the commands for configuring numbered ACL entries. The command to configure a numbered ACL is **access-list**. The command for configuring a named ACL is **ip access-list**. In addition, when you configure a numbered ACL entry, you specify all the command parameters on the same command. When you configure a named ACL, you specify the ACL type (standard or extended) and the ACL number with one command, which places you in the configuration level for that ACL. Once you enter the configuration level for the ACL, the command syntax is the same as the syntax for numbered ACLs.

Extended ACLs let you permit or deny packets based on the following information:

- IP protocol
- Source IP address or host name
- Destination IP address or host name
- Source TCP or UDP port (if the IP protocol is TCP or UDP)
- Destination TCP or UDP port (if the IP protocol is TCP or UDP)

The IP protocol can be one of the following well-known names or any IP protocol number from 0 – 255:

- Internet Control Message Protocol (ICMP)
- Internet Group Management Protocol (IGMP)
- Internet Gateway Routing Protocol (IGRP)
- Internet Protocol (IP)

- Open Shortest Path First (OSPF)
- Transmission Control Protocol (TCP)
- User Datagram Protocol (UDP)

For TCP and UDP, you also can specify a comparison operator and port name or number. For example, you can configure a policy to block web access to a specific website by denying all TCP port 80 (HTTP) packets from a specified source IP address to the website's IP address.

Extended Named ACL Syntax

Syntax: [no] ip access-list extended <acl-name>

deny | permit <ip-protocol> <source-ip> | <hostname> <wildcard> [<operator> <source-tcp/udp-port>] <destination-ip> | <hostname> [<icmp-num> | <icmp-type>] <wildcard> [<tcp/udp comparison operator> <destination-tcp/udp-port>] [dscp-cos-mapping] [dscp-marking <0-63> [802.1p-priority-marking <0 –7>... | dscp-cos-mapping]] [dscp-matching <0-63>] [log] [precedence <name> | <0 – 7>] [tos <0 – 63> | <name>] [traffic policy <name>]

Syntax: [no] access-list <num> deny | permit host <ip-protocol> any any

Syntax: [no] ip access-group <num> in

The <acl-name> parameter is the access list name. You can specify a string of up to 256 alphanumeric characters. You can use blanks in the ACL name if you enclose the name in quotation marks (for example, "ACL for Net1").

The **deny | permit** parameter indicates whether packets that match the policy are dropped or forwarded.

The <ip-protocol> parameter indicates the type of IP packet you are filtering. You can specify a well-known name for any protocol whose number is less than 255. For other protocols, you must enter the number. Enter "?" instead of a protocol to list the well-known names recognized by the CLI.

The <source-ip> | <hostname> parameter specifies the source IP host for the policy. If you want the policy to match on all source addresses, enter **any**.

The <wildcard> parameter specifies the portion of the source IP host address to match against. The <wildcard> is a four-part value in dotted-decimal notation (IP address format) consisting of ones and zeros. Zeros in the mask mean the packet's source address must match the <source-ip>. Ones mean any value matches. For example, the <source-ip> and <wildcard> values 209.157.22.26 0.0.0.255 mean that all hosts in the Class C subnet 209.157.22.x match the policy.

If you prefer to specify the wildcard (mask value) in Classless Interdomain Routing (CIDR) format, you can enter a forward slash after the IP address, then enter the number of significant bits in the mask. For example, you can enter the CIDR equivalent of "209.157.22.26 0.0.0.255" as "209.157.22.26/24". The CLI automatically converts the CIDR number into the appropriate ACL mask (where zeros instead of ones are the significant bits) and changes the non-significant portion of the IP address into zeros. For example, if you specify 209.157.22.26/24 or 209.157.22.26 0.0.0.255, then save the changes to the startup-config file, the value appears as 209.157.22.0/24 (if you have enabled display of subnet lengths) or 209.157.22.0 0.0.0.255 in the startup-config file.

If you enable the software to display IP subnet masks in CIDR format, the mask is saved in the file in "/<mask-bits>" format. To enable the software to display the CIDR masks, enter the **ip show-subnet-length** command at the global CONFIG level of the CLI. You can use the CIDR format to configure the ACL entry regardless of whether the software is configured to display the masks in CIDR format.

NOTE: If you use the CIDR format, the ACL entries appear in this format in the running-config and startup-config files, but are shown with subnet mask in the display produced by the **show ip access-list** command.

The <destination-ip> | <hostname> parameter specifies the destination IP host for the policy. If you want the policy to match on all destination addresses, enter **any**.

The <icmp-type> | <icmp-num> parameter specifies the ICMP protocol type.

- This parameter applies only if you specified **icmp** as the <ip-protocol> value.
- If you use this parameter, the ACL entry is sent to the CPU for processing.
- If you do not specify a message type, the ACL applies to all types of ICMP messages.

The <icmp-num> parameter can be a value from 0 – 255.

The <icmp-type> parameter can have one of the following values, depending on the software version the device is running:

- any-icmp-type
- echo
- echo-reply
- information-request
- log
- mask-reply
- mask-request
- parameter-problem
- redirect
- source-quench
- time-exceeded
- timestamp-reply
- timestamp-request
- traffic policy
- unreachable
- <num>

The <tcp/udp comparison operator> parameter specifies a comparison operator for the TCP or UDP port number. This parameter applies only when you specify **tcp** or **udp** as the IP protocol. For example, if you are configuring an entry for HTTP, specify **tcp eq http**. You can enter one of the following operators:

- **eq** – The policy applies to the TCP or UDP port name or number you enter after **eq**.
- **established** – This operator applies only to TCP packets. If you use this operator, the policy applies to TCP packets that have the ACK (Acknowledgment) or RST (Reset) bits set on (set to “1”) in the Control Bits field of the TCP packet header. Thus, the policy applies only to established TCP sessions, not to new sessions. See Section 3.1, “Header Format”, in RFC 793 for information about this field.

  NOTE: This operator applies only to destination TCP ports, not source TCP ports.

- **gt** – The policy applies to TCP or UDP port numbers greater than the port number or the numeric equivalent of the port name you enter after **gt**.
- **lt** – The policy applies to TCP or UDP port numbers that are less than the port number or the numeric equivalent of the port name you enter after **lt**.
- **neq** – The policy applies to all TCP or UDP port numbers except the port number or port name you enter after **neq**.

- **range** – The policy applies to all TCP or UDP port numbers that are between the first TCP or UDP port name or number and the second one you enter following the **range** parameter. The range includes the port names or numbers you enter. For example, to apply the policy to all ports between and including 23 (Telnet) and 53 (DNS), enter the following: **range 23 53**. The first port number in the range must be lower than the last number in the range.

The <tcp/udp-port> parameter specifies the TCP or UDP port number or well-known name. You can specify a well-known name for any application port whose number is less than 1024. For other application ports, you must enter the number. Enter "?" instead of a port to list the well-known names recognized by the CLI.

The **in** parameter specifies that the ACL applies to incoming traffic on the interface to which you apply the ACL. You can apply the ACL to an Ethernet port or a virtual interface.

NOTE: If the ACL is for a virtual routing interface, you also can specify a subset of ports within the VLAN containing that interface when assigning an ACL to the interface. See "Configuring Standard Numbered ACLs" on page 17-4.

The **precedence** <name> | <num> parameter of the **ip access-list** command specifies the IP precedence. The precedence option for of an IP packet is set in a three-bit field following the four-bit header-length field of the packet's header. You can specify one of the following:

- **critical** or **5** – The ACL matches packets that have the critical precedence. If you specify the option number instead of the name, specify number 5.
- **flash** or **3** – The ACL matches packets that have the flash precedence. If you specify the option number instead of the name, specify number 3.
- **flash-override** or **4** – The ACL matches packets that have the flash override precedence. If you specify the option number instead of the name, specify number 4.
- **immediate** or **2** – The ACL matches packets that have the immediate precedence. If you specify the option number instead of the name, specify number 2.
- **internet** or **6** – The ACL matches packets that have the internetwork control precedence. If you specify the option number instead of the name, specify number 6.
- **network** or **7** – The ACL matches packets that have the network control precedence. If you specify the option number instead of the name, specify number 7.
- **priority** or **1** – The ACL matches packets that have the priority precedence. If you specify the option number instead of the name, specify number 1.
- **routine** or **0** – The ACL matches packets that have the routine precedence. If you specify the option number instead of the name, specify number 0.

The **tos** <name> | <num> parameter of the **ip access-list** command specifies the IP ToS. You can specify one of the following:

- **max-reliability** or **2** – The ACL matches packets that have the maximum reliability ToS. The decimal value for this option is 2.
- **max-throughput** or **4** – The ACL matches packets that have the maximum throughput ToS. The decimal value for this option is 4.
- **min-delay** or **8** – The ACL matches packets that have the minimum delay ToS. The decimal value for this option is 8.
- **min-monetary-cost** or **1** – The ACL matches packets that have the minimum monetary cost ToS. The decimal value for this option is 1.

  NOTE: This value is not supported on 10 Gigabit Ethernet modules.

- **normal** or **0** – The ACL matches packets that have the normal ToS. The decimal value for this option is 0.

- <num> – A number from 0 – 15 that is the sum of the numeric values of the options you want. The ToS field is a four-bit field following the Precedence field in the IP header. You can specify one or more of the following. To select more than one option, enter the decimal value that is equivalent to the sum of the numeric values of all the ToS options you want to select. For example, to select the **max-reliability** and **min-delay** options, enter number 10. To select all options, select 15.

The **dscp-cos-mapping** option maps the DSCP value in incoming packets to a hardware table that provides mapping of each of the 0 – 63 DSCP values, and distributes them among eight traffic classes (internal priorities) and eight 802.1p priorities.

NOTE: The **dscp-cos-mapping** option overrides port-based priority settings.

The **dscp-marking** option enables you to configure an ACL that marks matching packets with a specified DSCP value Enter a value from 0 – 63. See “Using an IP ACL to Mark DSCP Values (DSCP Marking)” on page 17-29.

The **dscp-matching** option matches on the packet’s DSCP value. Enter a value from 0 – 63. This option does not change the packet’s forwarding priority through the device or mark the packet. See “DSCP Matching” on page 17-30.

The **log** parameter enables SNMP traps and Syslog messages for packets denied by the ACL.

You can enable logging on ACLs and filters that support logging even when the ACLs and filters are already in use. To do so, re-enter the ACL or filter command and add the **log** parameter to the end of the ACL or filter. The software replaces the ACL or filter command with the new one. The new ACL or filter, with logging enabled, takes effect immediately.

The **traffic-policy** option enables the device to rate limit inbound traffic and to count the packets and bytes per packet to which ACL permit or deny clauses are applied. For configuration procedures and examples, see the chapter “Configuring Traffic Policies” on page 24-1.

Configuration Example for Extended Named ACLs

To configure an extended named ACL, enter commands such as the following.

```
FastIron(config)#ip access-list extended “block Telnet”
FastIron(config-ext-nacl)#deny tcp host 209.157.22.26 any eq telnet log
FastIron(config-ext-nacl)#permit ip any any
FastIron(config-ext-nacl)#exit
FastIron(config)#int eth 1/1
FastIron(config-if-1/1)#ip access-group “block Telnet” in
```

The options at the ACL configuration level and the syntax for the **ip access-group** command are the same for numbered and named ACLs and are described in “Configuring Extended Numbered ACLs” on page 17-8 and “Configuring Extended Named ACLs” on page 17-13.

Preserving User Input for ACL TCP/UDP Port Numbers

Prior to release 03.1.00 for the FastIron X Series, ACL implementations automatically display the TCP/UDP port name instead of the port number, regardless of user preference. This feature preserves the user input (name or number) and now displays either the port name or the number.

A new command has been added to enable this feature:

```
FastIron(config)#ip preserve-acl-user-input-format
```

Syntax: ip preserve-acl-user-input-format

The following example shows how this feature works for a TCP port (this feature works the same way for UDP ports). In this example, the user identifies the TCP port by number (80) when configuring ACL group 140. However, **show ip access-list 140** reverts back to the port name for the TCP port (http in this example). After the user issues the new **ip preserve-acl-user-input-format** command, **show ip access-list 140** displays either the TCP port number or name, depending on how it was configured by the user.

```
FastIron(config)#access-list 140 permit tcp any any eq 80
FastIron(config)#access-list 140 permit tcp any any eq ftp
FastIron#show ip access-lists 140

Extended IP access list 140
permit tcp any any eq http
permit tcp any any eq ftp

FastIron(config)#ip preserve-acl-user-input-format
FastIron#show ip access-lists 140

Extended IP access list 140
permit tcp any any eq 80
permit tcp any any eq ftp
```

Adding a Comment to an ACL Entry

You can optionally add comment text to describe entries in an ACL. The comment text appears in the output of **show** commands that display ACL information.

For example, the following commands add comments to entries to a numbered ACL, ACL 100:

```
FastIron(config)#access-list 100 remark The following line permits TCP packets
FastIron(config)#access-list 100 permit tcp 192.168.4.40/24 2.2.2.2/24

FastIron(config)#access-list 100 remark The following permits UDP packets
FastIron(config)#access-list 100 permit udp 192.168.2.52/24 2.2.2.2/24
FastIron(config)#access-list 100 deny ip any any
```

If the ACL is a named ACL, (for example, you entered TCP/UDP instead of 100), enter the following commands:

```
FastIron(config)#access-list TCP/UDP remark The following line permits TCP packets
FastIron(config)#access-list TCP/UDP permit tcp 192.168.4.40/24 2.2.2.2/24
FastIron(config)#access-list TCP/UDP remark The following permits UDP packets
FastIron(config)#access-list TCP/UDP permit udp 192.168.2.52/24 2.2.2.2/24
FastIron(config)#access-list TCP/UDP deny ip any any
```

Syntax: [no] access-list <acl-num> | <acl-name> remark <comment-text>

Enter the number of the ACL for <acl-num>. You can add a comment to a named ACL by entering the ACL's name for <acl-name>.

The <comment-text> can be up to 128 characters in length. The comment must be entered separately from the actual ACL entry; that is, you cannot enter the ACL entry and the ACL comment with the same **access-list** command. Also, in order for the remark to be displayed correctly in the output of **show** commands, the comment must be entered immediately before the ACL entry it describes.

You can use the **show running-config** or **show access-list** commands to display the ACL and comments

The following shows an example of a numbered ACL with a comment text in a show running-config display:

```
FastIron#show running-config
…
access-list 100 remark The following line permits TCP packets
access-list 100 permit tcp 192.168.4.40/24 2.2.2.2/24
access-list 100 remark The following line permits UDP packets
access-list 100 permit udp 192.168.2.52/24 2.2.2.2/24
access-list 100 deny ip any any
```

The following shows the comment text for the ACL named TCP/UDP in a show running-config display:

```
FastIron#show running-config ...

access-list TCP/UDP remark The following line permits TCP packets
access-list TCP/UDP permit tcp 192.168.4.40/24 2.2.2.2/24
access-list TCP/UDP remark The following line permits UDP packets
access-list TCP/UDP permit udp 192.168.2.52/24 2.2.2.2/24
access-list TCP/UDP deny ip any any
```

Syntax: show running-config

The following example show the comment text for a numbered ACL in a show access-list display:

```
FastIron#show access-list 100
IP access list rate-limit 100 aaaa.bbbb.cccc

Extended IP access list 100 (Total flows: N/A, Total packets: N/A)
  ACL Comments:  The following line permits TCP packets
  permit tcp 0.0.0.40 255.255.255.0 0.0.0.2 255.255.255.0 (Flows: N/A, Packets: N/A)
  ACL Comments:  The following line permits UDP packets
  permit udp 0.0.0.52 255.255.255.0 0.0.0.2 255.255.255.0 (Flows: N/A, Packets: N/A)
  deny ip any any (Flows: N/A, Packets: N/A)
```

The next example shows the comment text for a named ACL in a show access-list display:

```
FastIron#show access-list TCP/UDP

IP access list rate-limit 100 aaaa.bbbb.cccc

Extended IP access list TCP/UDP (Total flows: N/A, Total packets: N/A)
ACL Comments:  The following line permits TCP packets
permit tcp 0.0.0.40 255.255.255.0 0.0.0.2 255.255.255.0 (Flows: N/A, Packets: N/A)
ACL Comments:  The following line permits UDP packets
permit udp 0.0.0.52 255.255.255.0 0.0.0.2 255.255.255.0 (Flows: N/A, Packets: N/A)
deny ip any any (Flows: N/A, Packets: N/A)
```

Syntax: show access-list <acl-num> | <acl-name> | all

Applying an ACL to a Virtual Interface in a Protocol- or Subnet-based VLAN

By default, when you apply an ACL to a virtual interface in a protocol-based or subnet-based VLAN, the ACL takes effect on all protocol or subnet VLANs to which the untagged port belongs. To prevent the Foundry device from denying packets on other virtual interfaces that do not have an ACL applied, configure an ACL that permits packets in the IP subnet of the virtual interface in all protocol-based or subnet-based VLANs to which the untagged port belongs. The following is an example configuration:

```
FastIron#conf t
FastIron(config)#vlan 1 name DEFAULT-VLAN by port
FastIron(config-vlan-1)#ip-subnet 192.168.10.0 255.255.255.0
FastIron(config-vlan-ip-subnet)#static ethe 1
FastIron(config-vlan-ip-subnet)#router-interface ve 10
FastIron(config-vlan-ip-subnet)#ip-subnet 10.15.1.0 255.255.255.0
FastIron(config-vlan-ip-subnet)#static ethe 1
FastIron(config-vlan-ip-subnet)#router-interface ve 20
FastIron(config-vlan-ip-subnet)#logging console
FastIron(config-vlan-ip-subnet)#exit
FastIron(config-vlan-1)#no vlan-dynamic-discovery
  Vlan dynamic discovery is disabled
FastIron(config-vlan-1)#int e 2
FastIron(config-if-e1000-2)#disable
FastIron(config-if-e1000-2)#interface ve 10
FastIron(config-vif-10)#ip address 192.168.10.254 255.255.255.0
FastIron(config-vif-10)#int ve 20
FastIron(config-vif-20)#ip access-group test1 in
FastIron(config-vif-20)#ip address 10.15.1.10 255.255.255.0
FastIron(config-vif-20)#exit
FastIron(config)#ip access-list extended test1
FastIron(config-ext-nacl)#permit ip 10.15.1.0 0.0.0.255 any log
FastIron(config-ext-nacl)#permit ip 192.168.10.0 0.0.0.255 any log
FastIron(config-ext-nacl)#end
FastIron#
```

Enabling ACL Logging

You may want the software to log entries in the Syslog for packets that are denied by ACL filters. ACL logging is disabled by default; it must be explicitly enabled on a port.

When you enable logging for ACL entries, statistics for packets that match the deny conditions of the ACL entries are logged. For example, if you configure a standard ACL entry to deny all packets from source address 209.157.22.26, statistics for packets that are explicitly denied by the ACL entry are logged in the Syslog buffer and in SNMP traps sent by the Foundry device.

The first time an ACL entry denies a packet, the software immediately generates a Syslog entry and an SNMP trap. The software also starts a five-minute timer. The timer keeps track of all packets explicitly denied by the ACL entries. After five minutes, the software generates a single Syslog entry for each ACL entry that denied a packet. The Syslog entry (message) indicates the number of packets denied by the ACL entry during the previous five minutes. Note however that packet count may be inaccurate if the packet rate is high and exceeds the CPU processing rate.

If no ACL entries explicitly deny packets during an entire five-minute timer interval, the timer stops. The timer restarts when an ACL entry explicitly denies a packet.

NOTE: The timer for logging packets denied by Layer 2 filters is different timer than the ACL logging timer.

Configuration Notes

Note the following before configuring ACL logging:

- You can enable ACL logging on physical and virtual interfaces.
- ACL logging logs denied packets only.
- When ACL logging is disabled, packets that match the ACL rule are forwarded or dropped in hardware. When ACL logging is enabled, all packets that match the ACL deny rule are sent to the CPU. When ACL

logging is enabled, Foundry recommends that you configure a traffic conditioner then link the ACL to the traffic conditioner to prevent CPU overload. For example:

```
FastIron(config)#traffic-policy TPD1 rate-limit fixed 100 exceed-action drop
FastIron(config)#access-list 101 deny ip host 210.10.12.2 any traffic-policy
TPD1 log
```

- ACL logging is intended for debugging purpose. Foundry recommends that you disable ACL logging after the debug session is over.

Enabling ACL Logging

To enable ACL logging, complete the following steps:

- Create ACL entries with the log option
- Enable ACL logging on individual ports

NOTE: The command syntax for enabling ACL logging is different on IPv4 devices than on IPv6 devices. See the configuration examples in the next section.

- Bind the ACLs to the ports on which ACL logging is enabled

Example Configurations

The following shows an example configuration on an IPv4 device.

```
FastIron(config)#access-list 1 deny host 209.157.22.26 log
FastIron(config)#access-list 1 deny 209.157.29.12 log
FastIron(config)#access-list 1 deny host IPHost1 log
FastIron(config)#access-list 1 permit any
FastIron(config)#interface e 1/4
FastIron(config-if-e1000-1/4)#acl-logging
FastIron(config-if-e1000-1/4)#ip access-group 1 in
```

The above commands create ACL entries that include the log option, enable ACL logging on interface e 1/4, then bind the ACL to interface e 1/4. Statistics for packets that match the deny statements will be logged.

NOTE: The **acl-logging** command shown above is **not** required for FGS and FLS devices.

Syntax: acl-logging

The **acl-logging** command applies to IPv4 devices only. For IPv6 devices, use the **logging-enable** command as shown in the following example.

The following shows an example configuration on an IPv6 device.

```
FastIron(config)#access-list 1 deny host 209.157.22.26 log
FastIron(config)#access-list 1 deny 209.157.29.12 log
FastIron(config)#access-list 1 deny host IPHost1 log
FastIron(config)#access-list 1 permit any
FastIron(config)#interface e 1/4
FastIron(config-if-e1000-1/4)#logging-enable
FastIron(config-if-e1000-1/4)#ip access-group 1 in
```

The above commands create ACL entries that include the log option, enable ACL logging on interface e 1/4, then bind the ACL to interface e 1/4. Statistics for packets that match the deny statements will be logged.

Syntax: logging-enable

The **logging-enabled** command applies to IPv6 devices only. For IPv4 devices, use the **acl-logging** command as shown in the previous example.

Displaying ACL Log Entries

The first time an entry in an ACL permits or denies a packet and logging is enabled for that entry, the software generates a Syslog message and an SNMP trap. Messages for packets permitted or denied by ACLs are at the warning level of the Syslog.

When the first Syslog entry for a packet permitted or denied by an ACL is generated, the software starts an ACL timer. After this, the software sends Syslog messages every five minutes. If an ACL entry does not permit or deny any packets during the timer interval, the software does not generate a Syslog entry for that ACL entry.

NOTE: For an ACL entry to be eligible to generate a Syslog entry for denied packets, logging must be enabled for the entry. The Syslog contains entries only for the ACL entries that deny packets and have logging enabled.

To display Syslog entries, enter the following command from any CLI prompt:

```
FastIron#show log
Syslog logging: enabled (0 messages dropped, 2 flushes, 0 overruns)
    Buffer logging: level ACDMEINW, 9 messages logged
    level code: A=alert C=critical D=debugging M=emergency E=error
                I=informational N=notification W=warning

Dynamic Log Buffer (50 lines):
0d00h12m18s:W:ACL: ACL: List 122 denied tcp 20.20.15.6(0)(Ethernet 4
0000.0804.0101) -> 20.20.18.6(0), 1 event(s)
0d00h12m18s:W:ACL: ACL: List 122 denied tcp 20.20.15.2(0)(Ethernet 4
0000.0804.0101) -> 20.20.18.2(0), 1 event(s)
0d00h12m18s:W:ACL: ACL: List 122 denied tcp 20.20.15.4(0)(Ethernet 4
0000.0804.0101) -> 20.20.18.4(0), 1 event(s)
0d00h12m18s:W:ACL: ACL: List 122 denied tcp 20.20.15.3(0)(Ethernet 4
0000.0804.0101) -> 20.20.18.3(0), 1 event(s)
0d00h12m18s:W:ACL: ACL: List 122 denied tcp 20.20.15.5(0)(Ethernet 4
0000.0804.0101) -> 20.20.18.5(0), 1 event(s)
0d00h12m18s:I:ACL: 122 applied to port 4 by  from console session
0d00h10m12s:I:ACL: 122 removed from port 4 by  from console session
0d00h09m56s:I:ACL: 122 removed from port 4 by  from console session
0d00h09m38s:I:ACL: 122 removed from port 4 by  from console session
```

Syntax: show log

Enabling Strict Control of ACL Filtering of Fragmented Packets

---

NOTE: This feature is not supported on the FastIron GS.

The default processing of fragments by hardware-based ACLs is as follows:

- The first fragment of a packet is permitted or denied using the ACLs. The first fragment is handled the same way as non-fragmented packets, since the first fragment contains the Layer 4 source and destination application port numbers. The device uses the Layer 4 CAM entry if one is programmed, or applies the interface's ACL entries to the packet and permits or denies the packet according to the first matching ACL.
- For other fragments of the same packet, they are subject to a rule only if there is no Layer 4 information in the rule or in any preceding rules.

The fragments are forwarded even if the first fragment, which contains the Layer 4 information, was denied. Generally, denying the first fragment of a packet is sufficient, since a transaction cannot be completed without the entire packet.

For tighter control, you can configure the port to drop all packet fragments. To do so, enter commands such as the following:

```
FastIron(config)#interface ethernet 1/1
FastIron(config-if-1/1)#ip access-group frag deny
```

This option begins dropping all fragments received by the port as soon as you enter the command. This option is especially useful if the port is receiving an unusually high rate of fragments, which can indicate a hacker attack.

Syntax: [no] ip access-group frag deny

Enabling ACL Support for Switched Traffic in the Router Image

Platform Support:

- FESX/FSX/FWSX devices running software release 02.5.00 and later
- FGS and FLS devices running software release 04.0.00 and later

By default, when an ACL is applied to a physical or virtual routing interface, the Foundry Layer 3 device filters routed traffic only. It does not filter traffic that is switched from one port to another within the same VLAN or virtual routing interface, even if an ACL is applied to the interface.

Starting in software release 02.5.00, you can enable the device to filter switched traffic within a VLAN or virtual routing interface. When filtering is enabled, the device uses the ACLs applied to inbound traffic to filter traffic received by a port from another port in the same virtual routing interface.

To enable this feature, enter a command such as the following:

```
FastIron(config)#access-list 101 bridged-routed
```

Applying the ACL rule above to an interface, enables filtering of traffic switched within a VLAN or virtual routing interface.

Syntax: [no] ip access-list <ACL-ID> bridged-routed

The <ACL-ID> parameter specifies a standard or extended numbered or named ACL.

You can use this feature in conjunction with **enable acl-per-port-per-vlan**, to assign an ACL to a single port within a virtual interface. In this case, all of the Layer 3 traffic (bridged and routed) are filtered by the ACL.

```
FastIron(config)#enable acl-per-port-per-vlan
FastIron(config)#write memory
FastIron(config)#exit
FastIron#reload
```

NOTE: You must save the configuration and reload the software to place the change into effect.

Enabling ACL Filtering Based on VLAN Membership or VE Port Membership

Platform Support:

- FESX/FSX/FWSX devices running software release 02.4.00 and later

You can apply an inbound ACL to specific VLAN members on a port (Layer 2 devices only) or to specific ports on a virtual interface (VE) (Layer 3 Devices only).

By default, this feature support is disabled. To enable it, enter the following commands at the Global CONFIG level of the CLI:

```
FastIron (config)#enable acl-per-port-per-vlan
FastIron (config)#write memory
FastIron (config)#exit
```

```
FastIron#reload
```

After entering the above commands, you can:

- Apply an ACL to specific VLAN members on a port – see “Applying an ACL to Specific VLAN Members on a Port (Layer 2 Devices Only)” on page 17-24
- Apply an ACL to a subset of ports on a VE – see “Applying an ACL to a Subset of Ports on a Virtual Interface (Layer 3 Devices Only)” on page 17-25

NOTE: You must save the configuration and reload the software to place the change into effect.

Syntax: [no] enable acl-per-port-per-vlan

Enter the **no** form of the command to disable this feature.

Configuration Notes

- Before enabling this feature on a FastIron IPv4 device, make sure the VLAN numbers are contiguous. For example, the VLAN numbers can be 201, 202, 203, and 204, but not 300, 401, 600, and 900. Note that this does not apply to FastIron IPv6 devices, which do support non-sequential VLAN numbers.
- On a FastIron IPv6 device, this feature works in conjunction with PIM or DVMRP on a virtual or loopback interface. However, the system cannot run PIM or DVMRP on a physical interface with an IP address configured on it, when this feature is enabled. Note that this applies to FastIron IPv6 devices only.
- On a FastIron IPv6 device, **acl-per-port-per-vlan** is supported on virtual interfaces, but not on routing interfaces.
- Foundry devices running software release 04.0.01 or later do not support a globally-configured PBR policy together with per-port-per-VLAN ACLs.

Applying an ACL to Specific VLAN Members on a Port (Layer 2 Devices Only)

Platform Support:

- FESX/FSX/FWSX devices running software release 02.3.03 and later

When you bind an ACL to a port, the port filters all inbound traffic on the port. However, on a tagged port, there may be a need to treat packets for one VLAN differently from packets for another VLAN. Starting with release 02.3.03, you can configure a tagged port on a Layer 2 device to filter packets based on the packets’ VLAN membership.

NOTE: Before you can bind an ACL to specific VLAN members on a port, you must first enable support for this feature. If this feature is not already enabled on your device, enable it as instructed in the section “Enabling ACL Filtering Based on VLAN Membership or VE Port Membership” on page 17-23.

To apply an ACL to a specific VLAN on a port, enter commands such as the following on a tagged port:

```
FastIron(config)#vlan 12 name vlan12
FastIron(config-vlan-12)#untag ethernet 5 to 8
FastIron(config-vlan-12)#tag ethernet 23 to 24
FastIron(config-vlan-12)#exit
FastIron(config)#access-list 10 deny host 209.157.22.26 log
FastIron(config)#access-list 10 deny 209.157.29.12 log
FastIron(config)#access-list 10 deny host IPHost1 log
FastIron(config)#access-list 10 permit
FastIron(config)#int e 1/23
FastIron(config-if-e1000-1/23))#per-vlan 12
FastIron(config-if-e1000-1/23-vlan-12))#ip access-group 10 in
```

The commands in this example configure port-based VLAN 12, and add ports e 5 – 8 as untagged ports and ports e 23 – 24 as tagged ports to the VLAN. The commands following the VLAN configuration commands configure ACL 10. Finally, the last three commands apply ACL 10 on VLAN 12 for which port e 23 is a member.

Syntax: per-vlan <VLAN ID>

Syntax: [no] ip access-group <ACL ID>

The <VLAN ID> parameter specifies the VLAN name or number to which you will bind the ACL.

The <ACL ID> parameter is the access list name or number.

Applying an ACL to a Subset of Ports on a Virtual Interface (Layer 3 Devices Only)

Platform Support:

- FESX/FSX/FWSX devices running software release 02.4.00 and later

You can apply an ACL to a virtual routing interface. The virtual interface is used for routing between VLANs and contains all the ports within the VLAN. The ACL applies to all the ports on the virtual routing interface. Starting with release 02.3.03, you also can specify a subset of ports within the VLAN containing a specified virtual interface when assigning an ACL to that virtual interface.

Use this feature when you do not want the ACLs to apply to all the ports in the virtual interface's VLAN or when you want to streamline ACL performance for the VLAN.

NOTE: Before you can bind an ACL to specific ports on a virtual interface, you must first enable support for this feature. If this feature is not already enabled on your device, enable it as instructed in the section "Enabling ACL Filtering Based on VLAN Membership or VE Port Membership" on page 17-23.

To apply an ACL to a subset of ports within a virtual interface, enter commands such as the following:

```
FastIron(config)#vlan 10 name IP-subnet-vlan
FastIron(config-vlan-10)#untag ethernet 1/1 to 2/12
FastIron(config-vlan-10)#router-interface ve 1
FastIron(config-vlan-10)#exit
FastIron(config)#access-list 1 deny host 209.157.22.26 log
FastIron(config)#access-list 1 deny 209.157.29.12 log
FastIron(config)#access-list 1 deny host IPHost1 log
FastIron(config)#access-list 1 permit any
FastIron(config)#interface ve 1/1
FastIron(config-vif-1/1)#ip access-group 1 in ethernet 1/1 ethernet 1/3 ethernet 2/
1 to 2/4
```

The commands in this example configure port-based VLAN 10, add ports 1/1 – 2/12 to the VLAN, and add virtual routing interface 1 to the VLAN. The commands following the VLAN configuration commands configure ACL 1. Finally, the last two commands apply ACL 1 to a subset of the ports associated with virtual interface 1.

Syntax: [no] ip access-group <ACL ID> in ethernet <stacknum>/<slotnum>/<portnum> [to<stacknum>/<slotnum>/<portnum>]

The <ACL ID> parameter is the access list name or number.

The <slotnum> parameter applies on chassis devices only. It does not apply on FESX devices.

Filtering on IP Precedence and ToS Values

To configure an extended IP ACL that matches based on IP precedence, enter commands such as the following:

```
FastIron(config)#access-list 103 deny tcp 209.157.21.0/24 209.157.22.0/24
precedence internet
FastIron(config)#access-list 103 deny tcp 209.157.21.0/24 eq ftp 209.157.22.0/24
precedence 6
FastIron(config)#access-list 103 permit ip any any
```

The first entry in this ACL denies TCP traffic from the 209.157.21.*x* network to the 209.157.22.x network, if the traffic has the IP precedence option “internet” (equivalent to “6”).

The second entry denies all FTP traffic from the 209.157.21.*x* network to the 209.157.22.x network, if the traffic has the IP precedence value “6” (equivalent to “internet”).

The third entry permits all packets that are not explicitly denied by the other entries. Without this entry, the ACL would deny all incoming or outgoing IP traffic on the ports to which you assign the ACL.

To configure an IP ACL that matches based on ToS, enter commands such as the following:

```
FastIron(config)#access-list 104 deny tcp 209.157.21.0/24 209.157.22.0/24 tos normal
FastIron(config)#access-list 104 deny tcp 209.157.21.0/24 eq ftp 209.157.22.0/24 tos 13
FastIron(config)#access-list 104 permit ip any any
```

The first entry in this IP ACL denies TCP traffic from the 209.157.21.*x* network to the 209.157.22.x network, if the traffic has the IP ToS option “normal” (equivalent to “0”).

The second entry denies all FTP traffic from the 209.157.21.*x* network to the 209.157.22.x network, if the traffic has the IP precedence value “13” (equivalent to “max-throughput”, “min-delay”, and “min-monetary-cost”).

The third entry permits all packets that are not explicitly denied by the other entries. Without this entry, the ACL would deny all incoming or outgoing IP traffic on the ports to which you assign the ACL.

TCP Flags - Edge Port Security

Platform Support:

- FGS and FLS devices running software release 02.6.00 and later

This release supports edge port security on FastIron GS and FastIron LS devices. This feature works in combination with IP ACL rules, and supports all 6 TCP flags present in the offset 13 of the TCP header:

- +|- urg = Urgent
- +|- ack = Acknowledge
- +|- psh = Push
- +|- rst = Reset
- +|- syn = Synchronize
- +|- fin = Finish

TCP flags can be combined with other ACL functions (such as dscp-marking and traffic policies), giving you greater flexibility when designing ACLs.

The TCP flags feature offers two options, match-all and match-any:

- **Match-any** - Indicates that incoming TCP traffic must be matched against any of the TCP flags configured as part of the match-any ACL rule. In CAM hardware, the number of ACL rules will match the number of configured flags.

- **Match-all** - Indicates that incoming TCP traffic must be matched against all of the TCP flags configured as part of the match-all ACL rule. In CAM hardware, there will be only one ACL rule for all configured flags. For example:

```
FastIron(config-ext-nacl)#permit tcp 1.1.1.1 0.0.0.255 eq 100 2.2.2.2 0.0.0.255 eq
300 match-all +urg +ack +syn -rst
```

This command configures a single rule in CAM hardware. This rule will contain all of the configured TCP flags (urg, ack, syn, and rst).

Using TCP Flags in Combination with Other ACL Features

The TCP Flags feature has the added capability of being combined with other ACL features. For example:

```
FastIron(config-ext-nacl)#permit tcp any any match-all +urg +ack +syn -rst traffic-
policy test
```

This command configures the ACL to match incoming traffic with the TCP Flags urg, ack, and syn and also to apply the traffic policy (rate, limit, etc.) to the matched traffic.

```
FastIron(config-ext-nacl)#permit tcp any any match-all +urg +ack +syn -rst tos
normal
```

This command configures the ACL to match incoming traffic with the flags urg, ack, and syn, and also sets the tos bit to normal when the traffic exits the device.

NOTE: TCP Flags combines the functionality of older features such as TCP Syn Attack and TCP Establish. Avoid configuring these older features on a port where you have configured TCP Flags. TCP Flags can perform all of the functions of TCP Syn Attack and TCP Establish, and more. However, if TCP Syn Attack is configured on a port along with TCP Flags, TCP Syn Attack will take precedence.

NOTE: If an ACL clause with match-any exists, and the system runs out of CAM, if the total number of TCP rules to TCP Flags will not fit within 1021 entries (the maximum rules allowed per device), then none of the TCP Flag rules will be programmed into the CAM hardware.

NOTE: If a range option and match-any TCP-flags are combined in the same ACL, the total number of rules will be calculated as: Total number of rules in CAM hardware = (number of rules for range)* (number of rules for match-any TCP-flags).

QoS Options for IP ACLs

Quality of Service (QoS) options enable you to perform QoS for packets that match the ACLs. Using an ACL to perform QoS is an alternative to directly setting the internal forwarding priority based on incoming port, VLAN membership, and so on. (This method is described in “Assigning QoS Priorities to Traffic” on page 21-4.)

The following QoS ACL options are supported:

- **dscp-cos-mapping** – This option is similar to the **dscp-matching** command (described below). This option maps the DSCP value in incoming packets to a hardware table that provides mapping of each of the 0 – 63 DSCP values, and distributes them among eight traffic classes (internal priorities) and eight 802.1p priorities.

  By default, the Foundry device does the *802.1p to CoS* mapping. If you want to change the priority mapping to *DSCP to CoS* mapping, you must enter the following ACL statement:

  ```
  permit ip any any dscp-cos-mapping
  ```

- **dscp-marking** – Marks the DSCP value in the outgoing packet with the value you specify.
  - **internal-priority-marking** and **802.1p-priority-marking** – Supported with the DSCP marking option, these commands assign traffic that matches the ACL to a hardware forwarding queue (**internal-priority-**

marking), and re-mark the packets that match the ACL with the 802.1p priority (**802.1p-priority-marking**).

- **dscp-matching** – Matches on the packet's DSCP value. This option does not change the packet's forwarding priority through the device or mark the packet.

Configuration Notes for the FastIron GS and FastIron LS

- The FastIron GS and FastIron LS do not support marking and prioritization simultaneously with the same rule. To achieve this, you need to create two separate rules. In other words, you can mark a rule with DSCP or 802.1p information, or you can prioritize a rule based on DSCP or 802.1p information. You can enable only one of the following ACL options per rule:
  - 802.1p-priority-marking
  - dscp-marking
  - internal-priority-marking

  For example, any one of the following commands is supported:

  ```
  FastIron(config)#access-list 101 permit ip any any dscp-marking 43
  ```

  or

  ```
  FastIron(config)#access-list 101 permit ip any any 802.1p-priority-marking
  ```

  or

  ```
  FastIron(config)#access-list 101 permit ip any any internal-priority-marking 6
  ```

  The following command is not supported:

  ```
  FastIron(config)#access-list 101 permit ip any any dscp-marking 43 802.1p-
  priority-marking 4 internal-priority-marking 6
  ```

Using an ACL to Map the DSCP Value (DSCP CoS Mapping)

The **dscp-cos-mapping** option maps the DSCP value in incoming packets to a hardware table that provides mapping of each of the 0 – 63 DSCP values, and distributes them among eight traffic classes (internal priorities) and eight 802.1p priorities.

NOTE: The **dscp-cos-mapping** option overrides port-based priority settings.

By default, the Foundry device does the *802.1p to CoS* mapping. If you want to change the priority mapping to *DSCP to CoS* mapping, you must enter the following ACL statement:

```
permit ip any any dscp-cos-mapping
```

The complete CLI syntax for this feature is shown in "Configuring Extended Numbered ACLs" on page 17-8 and "Configuring Extended Named ACLs" on page 17-13. The following shows the syntax specific to the DSCP Cos mapping feature.

Syntax: ... [dscp-marking <dscp-value> **dscp-cos-mapping**]

or

Syntax: ...[dscp-cos-mapping]

NOTE: The **dscp-cos-mapping** option should not be used when assigning an 802.1p priority. To assign an 802.1p priority to a specific DSCP (using **dscp-match**), re-assign the DSCP value as well. For example:

```
FastIron(config)#access-list 100 permit ip any any dscp-match 0 dscp-marking 0
802.1p 0 internal 1
```

Using an IP ACL to Mark DSCP Values (DSCP Marking)

The **dscp-marking** option for extended ACLs allows you to configure an ACL that marks matching packets with a specified DSCP value. You also can use DSCP marking to assign traffic to a specific hardware forwarding queue (see "Using an ACL to Change the Forwarding Queue" on page 17-30).

For example, the following commands configure an ACL that marks all IP packets with DSCP value 5. The ACL is then applied to incoming packets on interface 7. Consequently, all inbound packets on interface 7 are marked with the specified DSCP value.

```
FastIron(config)#access-list 120 permit ip any any dscp-marking 5 dscp-cos-mapping
FastIron(config)#interface 1/7
FastIron(config-if-e1000-1/7)#ip access-group 120 in
```

Syntax: ...**dscp-marking** <dscp-value>

The **dscp-marking** <dscp-value> parameter maps a DSCP value to an internal forwarding priority. The DSCP value can be from 0 – 63.

Combined ACL for 802.1p Marking

Platform Support:

- FESX/FSX/FWSX devices running software release 03.1.00 and later

Release 03.1.00a introduced a simple method for assigning an 802.1p priority value to packets without affecting the actual packet or the DSCP. Prior to this release, users were required to provide DSCP-marking and DSCP-matching information in order to assign 802.1p priority values, which required the deployment of a 64-line ACL to match all possible DSCP values. Users were also required to configure an internal priority marking value. This release allows users to easily specify 802.1p priority marking values directly, and changes internal priority marking from *required* to o*ptional*. If the user does not set a specific internal marking priority, the default value is the same as the 802.1p priority marking value. Priority values range from 0 to 7.

Two new ACL parameters support this feature, one required for priority marking and one optional for internal priority marking. These parameters apply to IP, and TCP, and UDP.

For IP

```
FastIron(config)#acc 104 per ip any any 802.1p-priority-marking 1
```

or the following command, which also assigns an optional internal-priority-marking value:

```
FastIron(config)#acc 104 per ip any any 802.1p-priority-marking 1 internal-priority-
marking 5
```

Syntax: access-list <num(100-199)> permit ip any any 802.1p-priority-marking <priority value (0-7)> [internal-priority-marking <value (0-7)>]

For TCP

```
FastIron(config)#acc 105 per tcp any any 802.1p-priority-marking 1
```

or the following command, which also assigns an optional internal-priority-marking value:

```
FastIron(config)#acc 105 per tcp any any 802.1p-priority-marking 1 internal-
priority-marking 5
```

Syntax: access-list <num(100-199)> permit tcp any any 802.1p-priority-marking <priority value (0-7)> [internal-priority-marking <value (0-7)>]

For UDP

```
FastIron(config) #acc 105 per udp any any 802.1p-priority-marking 1
```

or the following command, which also assigns an optional internal-priority-marking value:

```
FastIron(config) #acc 105 per udp any any 802.1p-priority-marking 1 internal-
priority-marking-3 5
```

Syntax: access-list <num(100-199)> permit udp any any 802.1p-priority-marking <priority value (0-7)> [internal-priority-marking-3 <value (0-7)>]

In each of these examples, in the first command the internal-priority value is not specified, which means it maintains a default value of 1 (equal to that of the 802.1p value). In the second command, the internal-priority value has been configured by the user to 5.

Using an ACL to Change the Forwarding Queue

The **802.1p-priority-marking** <0 – 7> parameter re-marks the packets of the 802.1Q traffic that match the ACL with this new 802.1p priority, or marks the packets of the non-802.1Q traffic that match the ACL with this 802.1p priority, later at the outgoing 802.1Q interface.

The 802.1p priority mapping is shown in Table 17.1.

The **internal-priority-marking** <0 – 7> parameter assigns traffic that matches the ACL to a specific hardware forwarding queue (qosp0 – qosp7>.

NOTE: The **internal-priority-marking** parameter overrides port-based priority settings.

In addition to changing the internal forwarding priority, if the outgoing interface is an 802.1Q interface, this parameter maps the specified priority to its equivalent 802.1p (CoS) priority and marks the packet with the new 802.1p priority. Table 17.1 lists the default mappings of hardware forwarding queues to 802.1p priorities on the FESX and FSX.

Table 17.1: Default Mapping of Forwarding Queues to 802.1p Priorities

Forwarding Queue	qosp0	qosp1	qosp2	qosp3	qosp4	qosp5	qosp6	qosp7
802.1p	0	1	2	3	4	5	6	7

The complete CLI syntax for 802.1p priority marking and internal priority marking is shown in “Configuring Extended Numbered ACLs” on page 17-8 and “Configuring Extended Named ACLs” on page 17-13. The following shows the syntax specific to these features.

Syntax: ... **dscp-marking** <0 – 63> **802.1p-priority-marking** <0 – 7> **internal-priority-marking** <0 – 7>]

DSCP Matching

The **dscp-matching** option matches on the packet’s DSCP value. This option does not change the packet’s forwarding priority through the device or mark the packet.

To configure an ACL that matches on a packet with DSCP value 29, enter a command such as the following:

```
FastIron(config)#access-list 112 permit ip 1.1.1.0 0.0.0.255 2.2.2.x 0.0.0.255 dscp-
matching 29
```

The complete CLI syntax for this feature is shown in “Configuring Extended Numbered ACLs” on page 17-8 and “Configuring Extended Named ACLs” on page 17-13. The following shows the syntax specific to this feature.

Syntax: ...**dscp-matching** <0 – 63>

NOTE: For complete syntax information, see “Extended Numbered ACL Syntax” on page 17-8.

ACL-Based Rate Limiting

Platform Support:

- FESX/FSX/FWSX devices running software release 02.3.03 and later

ACL-based rate limiting provides the facility to limit the rate for IP traffic that matches the permit conditions in extended IP ACLs. This feature is available in the Layer 2 and Layer 3 code.

For more details, including configuration procedures, see the chapter “Configuring Traffic Policies” on page 24-1.

ACL Counting

Platform Support:

- FESX/FSX/FWSX devices running software release 02.3.03 and later

ACL counting is a mechanism for counting the number of packets and the number of bytes per packet to which ACL filters are applied.

To see the configuration procedures for ACL counting, see “Configuring Traffic Policies” on page 24-1.

Using ACLs to Control Multicast Features

You can use ACLs to control the following multicast features:

- Limit the number of multicast groups that are covered by a static rendezvous point (RP)
- Control which multicast groups for which candidate RPs sends advertisement messages to bootstrap routers
- Identify which multicast group packets will be forwarded or blocked on an interface

For configuration procedures, see the chapter “Configuring IP Multicast Protocols” on page 30-1.

Displaying ACL Information

To display the number of Layer 4 CAM entries used by each ACL, enter the following command:

```
FastIron#show access-list all

Extended IP access list 100 (Total flows: N/A, Total packets: N/A, Total rule cam use: 3)
permit udp host 192.168.2.169 any (Flows: N/A, Packets: N/A, Rule cam use: 1)
permit icmp any any (Flows: N/A, Packets: N/A, Rule cam use: 1)
deny ip any any (Flows: N/A, Packets: N/A, Rule cam use: 1)
```

Syntax: show access-list <acl-num> | <acl-name> | all

The Rule cam use field lists the number of CAM entries used by the ACL or entry. The number of CAM entries listed for the ACL itself is the total of the CAM entries used by the ACL’s entries.

For flow-based ACLs, the Total flows and Flows fields list the number of Layer 4 session table flows in use for the ACL.

The Total packets and Packets fields apply only to flow-based ACLs.

Troubleshooting ACLs

Use the following methods to troubleshoot ACLs:

- To display the number of Layer 4 CAM entries being used by each ACL, enter the **show access-list** <acl-num> | <acl-name> | **all** command. See “Displaying ACL Information” on page 17-31.
- To determine whether the issue is specific to fragmentation, remove the Layer 4 information (TCP or UDP application ports) from the ACL, then reapply the ACL.

If you are using another feature that requires ACLs, either use the same ACL entries for filtering and for the other feature, or change to flow-based ACLs.

Policy-Based Routing (PBR)

Platform Support:

- FESX and FSX devices running software release 03.0.00 and later

Policy-Based Routing (PBR) allows you to use ACLs and route maps to selectively modify and route IP packets in hardware. The ACLs classify the traffic. Route maps that match on the ACLs set routing attributes for the traffic.

A PBR policy specifies the next hop for traffic that matches the policy. Using standard ACLs with PBR, you can route IP packets based on their source IP address. With extended ACLs, you can route IP packets based on all of the clauses in the extended ACL.

You can configure the Foundry device to perform the following types of PBR based on a packet's Layer 3 and Layer 4 information:

- Select the next-hop gateway.
- Send the packet to the null interface (null0).

When a PBR policy has multiple next hops to a destination, PBR selects the first live next hop specified in the policy that is up. If none of the policy's direct routes or next hops are available, the packet is routed in the normal way.

Configuration Considerations

- Policy-Based Routing is not supported in the Base Layer 3 code.
- Global Policy-Based Routing is not supported when IP Follow is configured on an interface.
- Starting in software release FSX 04.0.01, global Policy-Based Routing is not supported with per-port-per-VLAN ACLs.
- A PBR policy on an interface takes precedence over a global PBR policy.
- You cannot apply PBR on a port if that port already has ACLs, ACL-based rate limiting, DSCP-based QoS, MAC filtering.
- The number of route maps that you can define is limited by the available system memory, which is determined by the system configuration and how much memory other features use. When a route map is used in a PBR policy, the PBR policy uses up to six instances of a route map, up to five ACLs in a matching policy of each route map instance, and up to six next hops in a set policy of each route map instance. Note that the CLI will allow you configure more than six next hops in a route map; however, the extra next hops will not be placed in the PBR database. The route map could be used by other features like BGP or OSPF, which may use more than six next hops.
- ACLs with the **log** option configured should not be used for PBR purposes.
- PBR ignores explicit or implicit **deny ip any any** ACL entries, to ensure that for route maps that use multiple ACLs, the traffic is compared to all the ACLs. PBR also ignores any deny clauses in an ACL. Traffic that matches a deny clause is routed normally using Layer 3 paths.
- PBR always selects the first next hop from the next hop list that is up. If a PBR policy's next hop goes down, the policy uses another next hop if available. If no next hops are available, the device routes the traffic in the normal way.
- PBR is not supported for fragmented packets. If the PBR's ACL filters on Layer 4 information like TCP/UDP ports, fragmented packed are routed normally.
- You can change route maps or ACL definitions dynamically and do not need to rebind the PBR policy to an interface.

Configuring a PBR Policy

To configure PBR, you define the policies using IP ACLs and route maps, then enable PBR globally or on individual interfaces. The device programs the ACLs into the packet processor on the interfaces and routes traffic that matches the ACLs according to the instructions in the route maps.

To configure a PBR policy:

- Configure ACLs that contain the source IP addresses for the IP traffic you want to route using PBR.
- Configure a route map that matches on the ACLs and sets the route information.
- Apply the route map to an interface.

Configure the ACLs

PBR uses route maps to change the routing attributes in IP traffic. This section shows an example of how to configure a standard ACL to identify the source subnet for IP traffic.

To configure a standard ACL to identify a source subnet, enter a command such as the following:

```
FastIron(config)#access-list 99 permit 209.157.23.0 0.0.0.255
```

The command in this example configures a standard ACL that permits traffic from subnet 209.157.23.0/24. After you configure a route map that matches based on this ACL, the software uses the route map to set route attributes for the traffic, thus enforcing PBR.

NOTE: Do not use an access group to apply the ACL to an interface. Instead, use a route map to apply the ACL globally or to individual interfaces for PBR, as shown in the following sections.

Syntax: [no] access-list <num> deny | permit <source-ip> | <hostname> <wildcard>

or

Syntax: [no] access-list <num> deny | permit <source-ip>/<mask-bits> | <hostname>

Syntax: [no] access-list <num> deny | permit host <source-ip> | <hostname>

Syntax: [no] access-list <num> deny | permit any

The <num> parameter is the access list number and can be from 1 – 99.

The **deny | permit** parameter indicates whether packets that match a policy in the access list are denied (dropped) or permitted (forwarded).

NOTE: If you are configuring the ACL for use in a route map, always specify **permit**. Otherwise, the Foundry device will ignore deny clauses and packets that match deny clauses are routed normally.

The <source-ip> parameter specifies the source IP address. Alternatively, you can specify the host name.

NOTE: To specify the host name instead of the IP address, the host name must be configured using the Foundry device's DNS resolver. To configure the DNS resolver name, use the **ip dns server-address…** command at the global CONFIG level of the CLI.

The <wildcard> parameter specifies the mask value to compare against the host address specified by the <source-ip> parameter. The <wildcard> is a four-part value in dotted-decimal notation (IP address format) consisting of ones and zeros. Zeros in the mask mean the packet's source address must match the <source-ip>. Ones mean any value matches. For example, the <source-ip> and <wildcard> values 209.157.22.26 0.0.0.255 mean that all hosts in the Class C subnet 209.157.22.x match the policy.

If you prefer to specify the wildcard (mask value) in CIDR format, you can enter a forward slash after the IP address, then enter the number of significant bits in the mask. For example, you can enter the CIDR equivalent of "209.157.22.26 0.0.0.255" as "209.157.22.26/24". The CLI automatically converts the CIDR number into the appropriate ACL mask (where zeros instead of ones are the significant bits) and changes the non-significant portion of the IP address into zeros. For example, if you specify 209.157.22.26/24 or 209.157.22.26 0.0.0.255, then save the changes to the startup-config file, the value appears as 209.157.22.0/24 (if you have enabled display of subnet lengths) or 209.157.22.0 0.0.0.255 in the startup-config file.

If you enable the software to display IP subnet masks in CIDR format, the mask is saved in the file in "/<mask-bits>" format. To enable the software to display the CIDR masks, enter the **ip show-subnet-length** command at the global CONFIG level of the CLI. You can use the CIDR format to configure the ACL entry regardless of whether the software is configured to display the masks in CIDR format.

NOTE: If you use the CIDR format, the ACL entries appear in this format in the running-config and startup-config files, but are shown with subnet mask in the display produced by the **show ip access-list** command.

The **host** <source-ip> | <hostname> parameter lets you specify a host IP address or name. When you use this parameter, you do not need to specify the mask. A mask of all zeros (0.0.0.0) is implied.

The **any** parameter configures the policy to match on all host addresses.

NOTE: Do not use the **log** option in ACLs that will be used for PBR.

Configure the Route Map

After you configure the ACLs, you can configure a PBR route map that matches based on the ACLs and sets routing information in the IP traffic.

NOTE: The match and set statements described in this section are the only route-map statements supported for PBR. Other route-map statements described in the documentation apply only to the protocols with which they are described.

To configure a PBR route map, enter commands such as the following:

```
FastIron(config)#route-map test-route permit 99
FastIron(config-routemap test-route)#match ip address 99
FastIron(config-routemap test-route)#set ip next-hop 192.168.2.1
FastIron(config-routemap test-route)#exit
```

The commands in this example configure an entry in a route map named "test-route". The **match** statement matches on IP information in ACL 99. The **set** statement changes the next-hop IP address for packets that match to 192.168.2.1.

Syntax: [no] route-map <map-name> permit | deny <num>

The <map-name> is a string of characters that names the map. Map names can be up to 32 characters in length. You can define an unlimited number of route maps on the Foundry device, as long as system memory is available.

The **permit | deny** parameter specifies the action the Foundry device will take if a route matches a match statement.

- If you specify **deny**, the Foundry device does not apply a PBR policy to packets that match the ACLs in a match clause. Those packets are routed normally,
- If you specify **permit**, the Foundry device applies the match and set statements associated with this route map instance.

The <num> parameter specifies the instance of the route map you are defining. Routes are compared to the instances in ascending numerical order. For example, a route is compared to instance 1, then instance 2, and so on.

PBR uses up to six route map instances for comparison and ignores the rest.

Syntax: [no] match ip address <ACL-num-or-name>

The <ACL-num> parameter specifies a standard or extended ACL number or name.

Syntax: [no] set ip next hop <ip-addr>

This command sets the next-hop IP address for traffic that matches a match statement in the route map.

Syntax: [no] set interface null0

This command sends the traffic to the null0 interface, which is the same as dropping the traffic.

Enabling PBR

After you configure the ACLs and route map entries, you can enable PBR globally, on individual interfaces, or both as described in this section. To enable PBR, you apply a route map you have configured for PBR globally or locally.

Enabling PBR Globally

To enable PBR globally, enter a command such as the following at the global CONFIG level:

```
FastIron(config)#ip policy route-map test-route
```

This command applies a route map named “test-route” to all interfaces on the device for PBR.

Syntax: ip policy route-map <map-name>

Enabling PBR Locally

To enable PBR locally, enter commands such as the following:

```
FastIron(config)#interface ve 1
FastIron(config-vif-1)#ip policy route-map test-route
```

The commands in this example change the CLI to the Interface level for virtual interface 1, then apply the “test-route” route map to the interface. You can apply a PBR route map to Ethernet ports or virtual interfaces.

Syntax: ip policy route-map <map-name>

Enter the name of the route map you want to use for the route-map <map-name> parameter.

Configuration Examples

This section presents configuration examples for:

- “Basic Example” on page 17-36
- “Setting the Next Hop” on page 17-36
- “Setting the Output Interface to the Null Interface” on page 17-37
- “Trunk Formation” on page 17-37

Basic Example

The following commands configure and apply a PBR policy that routes HTTP traffic received on virtual routing interface 1 from the 10.10.10.x/24 network to 5.5.5.x/24 through next-hop IP address 1.1.1.1/24 or, if 1.1.1.x is unavailable, through 2.2.2.1/24.

```
FastIron(config)#access-list 101 permit tcp 10.10.10.0 0.0.0.255 eq http 5.5.5.0
0.0.0.255
FastIron(config)#route-map net10web permit 101
FastIron(config-routemap net10web)#match ip address 101
FastIron(config-routemap net10web)#set ip next-hop 1.1.1.1
FastIron(config-routemap net10web)#set ip next-hop 2.2.2.2
FastIron(config-routemap net10web)#exit
FastIron(config)#vlan 10
FastIron(config-vlan-10)#tagged ethernet 1/1 to 1/4

FastIron(config-vlan-10)#router-interface ve 1
FastIron(config)#interface ve 1
FastIron(config-vif-1)#ip policy route-map net10web
```

Syntax: [no] route-map <map-name> permit | deny <num>

Syntax: [no] set ip next hop <ip-addr>

This command sets the next-hop IP address for traffic that matches a match statement in the route map.

Setting the Next Hop

The following commands configure the Foundry device to apply PBR to traffic from IP subnets 209.157.23.x, 209.157.24.x, and 209.157.25.x. In this example, route maps specify the next-hop gateway for packets from each of these subnets.

- Packets from 209.157.23.x are sent to 192.168.2.1.
- Packets from 209.157.24.x are sent to 192.168.2.2.
- Packets from 209.157.25.x are sent to 192.168.2.3.

The following commands configure three standard ACLs. Each ACL contains one of the ACLs listed above. Make sure you specify **permit** instead of deny in the ACLs, so that the Foundry device permits the traffic that matches the ACLs to be further evaluated by the route map. If you specify **deny**, the traffic that matches the deny statements are routed normally. Notice that these ACLs specify **any** for the destination address.

```
FastIron(config)#access-list 50 permit 209.157.23.0 0.0.0.255
FastIron(config)#access-list 51 permit 209.157.24.0 0.0.0.255
FastIron(config)#access-list 52 permit 209.157.25.0 0.0.0.255
```

The following commands configure three entries in a route map called “test-route”. The first entry (permit 50) matches on the IP address information in ACL 50 above. For IP traffic from subnet 209.157.23.0/24, this route map entry sets the next-hop IP address to 192.168.2.1.

```
FastIron(config)#route-map test-route permit 50
FastIron(config-routemap test-route)#match ip address 50
FastIron(config-routemap test-route)#set ip next-hop 192.168.2.1
FastIron(config-routemap test-route)#exit
```

The following commands configure the second entry in the route map. This entry (permit 51) matches on the IP address information in ACL 51 above. For IP traffic from subnet 209.157.24.0/24, this route map entry sets the next-hop IP address to 192.168.2.2.

```
FastIron(config)#route-map test-route permit 51
FastIron(config-routemap test-route)#match ip address 51
FastIron(config-routemap test-route)#set ip next-hop 192.168.2.2
FastIron(config-routemap test-route)#exit
```

The following commands configure the third entry in the test-route route map. This entry (permit 52) matches on the IP address information in ACL 52 above. For IP traffic from subnet 209.157.25.0/24, this route map entry sets the next-hop IP address to 192.168.2.3.

```
FastIron(config)#route-map test-route permit 52
FastIron(config-routemap test-route)#match ip address 52
FastIron(config-routemap test-route)#set ip next-hop 192.168.2.3
FastIron(config-routemap test-route)#exit
```

The following command enables PBR by globally applying the test-route route map to all interfaces.

```
FastIron(config)#ip policy route-map test-route
```

Alternatively, you can enable PBR on specific interfaces, as shown in the following example. The commands in this example configure IP addresses in the three source subnets identified in ACLs 50, 51, and 52, then apply route map test-route to the interface.

```
FastIron(config)#interface ve 1
FastIron(config-vif-1)#ip address 209.157.23.1/24
FastIron(config-vif-1)#ip address 209.157.24.1/24
FastIron(config-vif-1)#ip address 209.157.25.1/24
FastIron(config-vif-1)#ip policy route-map test-route
```

Setting the Output Interface to the Null Interface

The following commands configure a PBR policy to send all traffic from 192.168.1.204/32 to the null interface, thus dropping the traffic instead of forwarding it.

```
FastIron(config)#access-list 56 permit 209.168.1.204 0.0.0.0
```

The following commands configure an entry in a route map called “file-13”. The first entry (permit 56) matches on the IP address information in ACL 56 above. For IP traffic from the host 209.168.1.204/32, this route map entry sends the traffic to the null interface instead of forwarding it, thus sparing the rest of the network the unwanted traffic.

```
FastIron(config)#route-map file-13 permit 56
FastIron(config-routemap file-13)#match ip address 56
FastIron(config-routemap file-13)#set interface null0
FastIron(config-routemap file-13)#exit
```

The following command enables PBR by globally applying the route map to all interfaces.

```
FastIron(config)#ip policy route-map file-13
```

Alternatively, you can enable PBR on specific interfaces, as shown in the following example. The commands in this example configure IP addresses in the source subnet identified in ACL 56, then apply route map file-13 to the interface.

```
FastIron(config)#interface ethernet 3/11
FastIron(config-if-e10000-3/11)#ip address 192.168.1.204/32
FastIron(config-if-e10000-3/11)#ip policy route-map file-13
```

Trunk Formation

When a trunk is formed, the PBR policy on the primary port applies to all the secondary ports. If a different PBR policy exists on a secondary port at the time of a trunk formation, that policy is overridden by the PBR policy on the primary port. If the primary port does not have a PBR policy, then the secondary ports will not have a PBR policy.

When a trunk is removed, the PBR policy that was applied to the trunk interface is unbound (removed) from former secondary ports. If global PBR is configured, the secondary ports adhere to the global PBR; otherwise, no PBR policy is bound to former secondary ports.

Chapter 18

Configuring IPv6 Access Control Lists (ACLs)

This chapter describes how Access Control Lists (ACLs) are implemented and configured on a Foundry IPv6 Layer 3 Switch.

Platform Support:

- FESX and FSX IPv6 devices running software release 04.1.00 and later – L2, BL3, L3

ACL Overview

Foundry supports IPv6 Access Control Lists (ACLs), which you can use for traffic filtering. You can configure up to 100 IPv6 ACLs.

An IPv6 ACL is composed of one or more conditional statements that pose an action (permit or deny) if a packet matches a specified source or destination prefix. There can be up to 1024 statements per device. This includes IPv6, IPv4, MAC filter routes, and default statements. When the maximum number of ACL rules are reached on the device, an error message will display on the console.

In ACLs with multiple statements, you can specify a priority for each statement.The specified priority determines the order in which the statement appears in the ACL. The last statement in each IPv6 ACL is an implicit deny statement for all packets that do not match the previous statements in the ACL.

You can configure an IPv6 ACL on a global basis, then apply it to the incoming IPv6 packets on specified interfaces. You can apply only one IPv6 ACL to an interface's incoming traffic. When an interface receives an IPv6 packet, it applies the statement(s) within the ACL in their order of appearance to the packet. As soon as a match occurs, the Foundry device takes the specified action (permit or deny the packet) and stops further comparison for that packet.

NOTE: IPv6 ACLs are supported on inbound traffic and are implemented in hardware, making it possible for the Foundry device to filter traffic at line-rate speed on 10 Gigabit interfaces.

Foundry's IPv6 ACLs enable traffic filtering based on the following information:

- IPv6 protocol
- Source IPv6 address
- Destination IPv6 address
- IPv6 message type
- Source TCP or UDP port (if the IPv6 protocol is TCP or UDP)
- Destination TCP or UDP port (if the IPv6 protocol is TCP or UDP)

The IPv6 protocol can be one of the following well-known names or any IPv6 protocol number from 0 – 255:

- Authentication Header (AHP)
- Encapsulating Security Payload (ESP)
- Internet Control Message Protocol (ICMP)
- Internet Protocol Version 6 (IPv6)
- Stream Control Transmission Protocol (SCTP)
- Transmission Control Protocol (TCP)
- User Datagram Protocol (UDP)

NOTE: TCP and UDP filters will be matched only if they are listed as the first option in the extension header.

For TCP and UDP, you also can specify a comparison operator and port name or number. For example, you can configure a policy to block web access to a specific website by denying all TCP port 80 (HTTP) packets from a specified source IPv6 address to the website's IPv6 address.

IPv6 ACLs also provide support for filtering packets based on DSCP.

This chapter contains the following sections:

- "Using IPv6 ACLs as Input to Other Features" on page 18-2
- "Configuring an IPv6 ACL" on page 18-2
- "Applying an IPv6 ACL to an Interface" on page 18-11
- "Displaying ACLs" on page 18-11

Configuration Notes

- IPv4 ACLs that filter based on VLAN membership or VE port membership (acl-per-port-per-VLAN), are supported together with IPv6 ACLs on the same device, as long as they are not bound to the same port or virtual interface.
- IPv4 source guard and IPv6 ACLs are supported together on the same device, as long as they are not configured on the same port or virtual Interface.

Using IPv6 ACLs as Input to Other Features

You can use an IPv6 ACL to provide input to other features such as route maps and distribution lists. When you use an ACL this way, use permit statements in the ACL to specify the traffic that you want to send to the other feature. If you use deny statements, the traffic specified by the deny statements is not supplied to the other feature.

Configuring an IPv6 ACL

To configure an IPv6 ACL, do the following:

- Create the ACL
- Apply the ACL to an interface

Example Configurations

To configure an access list that blocks all Telnet traffic received on port 1/1 from IPv6 host 2000:2382:e0bb::2, enter the following commands.

```
FastIron(config)# ipv6 access-list fdry
FastIron(config-ipv6-access-list-fdry)# deny tcp host 2000:2382:e0bb::2 any eq
telnet
FastIron(config-ipv6-access-list-fdry)# permit ipv6 any any
FastIron(config-ipv6-access-list-fdry)# exit
FastIron(config)# int eth 1/1
FastIron(config-if-1/1)# ipv6 traffic-filter fdry in
FastIron(config)# write memory
```

Here is another example of commands for configuring an ACL and applying it to an interface.

```
FastIron(config)# ipv6 access-list netw
FastIron(config-ipv6-access-list-netw)# permit icmp 2000:2383:e0bb::/64
2001:3782::/64
FastIron(config-ipv6-access-list-netw)# deny ipv6 host 2000:2383:e0ac::2 host
2000:2383:e0aa:0::24
FastIron(config-ipv6-access-list-netw)# deny udp any any
FastIron(config-ipv6-access-list-netw)# permit ipv6 any any
```

The first condition permits ICMP traffic from hosts in the 2000:2383:e0bb::*x* network to hosts in the 2001:3782::*x* network.

The second condition denies all IPv6 traffic from host 2000:2383:e0ac::2 to host 2000:2383:e0aa:0::24.

The third condition denies all UDP traffic.

The fourth condition permits all packets that are not explicitly denied by the other entries. Without this entry, the ACL would deny all incoming IPv6 traffic on the ports to which you assigned the ACL.

The following commands apply the ACL "netw" to the incoming traffic on port 1/2 and to the incoming traffic on port 4/3.

```
FastIron(config)# int eth 1/2
FastIron(config-if-1/2)# ipv6 traffic-filter netw in
FastIron(config-if-1/2)# exit
FastIron(config)# int eth 4/3
FastIron(config-if-4/3)# ipv6 traffic-filter netw in
FastIron(config)# write memory
```

Here is another example:

```
FastIron(config)# ipv6 access-list nextone
FastIron(config-ipv6-access-list rtr)# deny tcp 2001:1570:21::/24
2001:1570:22::/24
FastIron(config-ipv6-access-list rtr)# deny udp any range 5 6 2001:1570:22::/24
FastIron(config-ipv6-access-list rtr)# permit ipv6 any any
FastIron(config-ipv6-access-list rtr)# write memory
```

The first condition in this ACL denies TCP traffic from the 2001:1570:21::*x* network to the 2001:1570:22::*x* network.

The next condition denies UDP packets from any source with source UDP port in ranges 5 to 6 and whose destination is to the 2001:1570:22::/24 network.

The third condition permits all packets containing source and destination addresses that are not explicitly denied by the first two. Without this entry, the ACL would deny all incoming IPv6 traffic on the ports to which you assign the ACL.

A **show running-config** command displays the following:

```
 FastIron(config)# show running-config

ipv6 access-list rtr
 deny tcp 2001:1570:21::/24 2001:1570:22::/24
 deny udp any range rje 6 2001:1570:22::/24
 permit ipv6 any any
```

A **show ipv6 access-list** command displays the following:

```
 FastIron(config)# sh ipv6 access-list rtr

ipv6 access-list rtr: 3 entries
 10: deny tcp 2001:1570:21::/24 2001:1570:22::/24
 20: deny udp any range rje 6 2001:1570:22::/24
 30: permit ipv6 any any
```

The following commands apply the ACL “rtr” to the incoming traffic on ports 2/1 and 2/2.

```
FastIron(config)# int eth 2/1
FastIron(config-if-2/1)# ipv6 traffic-filter rtr in
FastIron(config-if-2/1)# exit
FastIron(config)# int eth 2/2
FastIron(config-if-2/2)# ipv6 traffic-filter rtr in
FastIron(config)# write memory
```

Default and Implicit IPv6 ACL Action

The default action when no IPv6 ACLs are configured on an interface is to permit all IPv6 traffic. However, once you configure an IPv6 ACL and apply it to an interface, the default action for that interface is to deny all IPv6 traffic that is not explicitly permitted on the interface.

- If you want to tightly control access, configure ACLs consisting of permit entries for the access you want to permit. The ACLs implicitly deny all other access.
- If you want to secure access in environments with many users, you might want to configure ACLs that consist of explicit deny entries, then add an entry to permit all access to the end of each ACL. The permit entry permits packets that are not denied by the deny entries.

Every IPv6 ACL has the following implicit conditions as its last match conditions:

1. **permit icmp any any nd-na** – Allows ICMP neighbor discovery acknowledgement.
2. **permit icmp any any nd-ns** – Allows ICMP neighbor discovery solicitation.
3. **deny ipv6 any any** – Denies IPv6 traffic. You must enter a **permit ipv6 any any** as the last statement in the access-list if you want to permit IPv6 traffic that were not denied by the previous statements.

The conditions are applied in the order shown above, with **deny ipv6 any any** as the last condition applied.

For example, if you want to deny ICMP neighbor discovery acknowledgement, then permit any remaining IPv6 traffic, enter commands such as the following:

```
FastIron(config)# ipv6 access-list netw
FastIron(config-ipv6-access-list-netw)# permit icmp 2000:2383:e0bb::/64
2001:3782::/64
FastIron(config-ipv6-access-list-netw)# deny icmp any any nd-na
FastIron(config-ipv6-access-list-netw)# permit ipv6 any any
```

The first permit statement permits ICMP traffic from hosts in the 2000:2383:e0bb::*x* network to hosts in the 2001:3782::*x* network.

The deny statement denies ICMP neighbor discovery acknowledgement.

The last entry permits all packets that are not explicitly denied by the other entries. Without this entry, the ACL will deny all incoming IPv6 traffic on the ports to which you assigned the ACL.

Furthermore, if you add the statement **deny icmp any any** in the access list, then all neighbor discovery messages will be denied. You must explicitly enter the **permit icmp any any nd-na** and **permit icmp any any nd-ns** statements just before the **deny icmp** statement if you want the ACLs to permit neighbor discovery as in the example below.

```
FastIron(config)# ipv6 access-list netw
FastIron(config-ipv6-access-list-netw)# permit icmp 2000:2383:e0bb::/64
2001:3782::/64
FastIron(config-ipv6-access-list-netw)# permit icmp any any nd-na
FastIron(config-ipv6-access-list-netw)# permit icmp any any nd-ns
FastIron(config-ipv6-access-list-netw)# deny icmp any any
FastIron(config-ipv6-access-list-netw)# permit ipv6 any any
```

ACL Syntax

NOTE: The following features are not supported:

- **ipv6-operator flow-label**
- **ipv6-operator fragments** when any protocol is specified. The option "fragments" can be specified only when "permit/deny ipv6" is specified. If you specify "tcp" or any other protocol instead of "ipv6" the keyword, "fragments" cannot be used.
- **ipv6-operator routing** when any protocol is specified. (Same limitation as for **ipv6-operator fragments)**

When creating ACLs, use the appropriate syntax below for the protocol you are filtering.

For IPv6 and Supported Protocols Other than ICMP, TCP, or UDP

Syntax: [no] ipv6 access-list <acl name>

Syntax: permit | deny <protocol>
<ipv6-source-prefix/prefix-length> | any | host <source-ipv6_address>
<ipv6-destination-prefix/prefix-length> | any | host <ipv6-destination-address>
[ipv6-operator [<value>]]
[802.1p-priority-matching <number>]
[dscp-marking <number> 802.1p-priority-marking <number> internal-priority-marking <number>] | [dscp-marking <dscp-value> dscp-cos-mapping] | [dscp-cos-mapping]

For ICMP

Syntax: [no] ipv6 access-list <acl name>

Syntax: permit | deny icmp <ipv6-source-prefix/prefix-length> | any | host <source-ipv6_address>
<ipv6-destination-prefix/prefix-length> | any | host <ipv6-destination-address>
[ipv6-operator [<value>]]
[[<icmp-type>][<icmp-code>]] | [<icmp-message>]
[802.1p-priority-matching <number>]
[dscp-marking <number> 802.1p-priority-marking <number> internal-priority-marking <number>]
[dscp-marking <dscp-value> dscp-cos-mapping]
[dscp-cos-mapping]

For TCP

Syntax: [no] ipv6 access-list <acl name>

Syntax: permit | deny <tcp>
<ipv6-source-prefix/prefix-length> | any | host <source-ipv6_address> [tcp-udp-operator [source-port-number]]
<ipv6-destination-prefix/prefix-length> | any | host <ipv6-destination-address> [tcp-udp-operator [destination-port- number]]
[ipv6-operator [<value>]]
[802.1p-priority-matching <number>]
[dscp-marking <number> 802.1p-priority-marking <number> internal-priority-marking <number>]
[dscp-marking <dscp-value> dscp-cos-mapping]
[dscp-cos-mapping]

For UDP

Syntax: [no] ipv6 access-list <acl name>

Syntax: permit | deny <udp>
<ipv6-source-prefix/prefix-length> | any | host <source-ipv6_address> [tcp-udp-operator [source port number]]
<ipv6-destination-prefix/prefix-length> | any | host <ipv6-destination-address> [tcp-udp-operator [destination port number]]
[ipv6-operator [<value>]]
[802.1p-priority-matching <number>]
[dscp-marking <number> 802.1p-priority-marking <number> internal-priority-marking <number>]
[dscp-marking <dscp-value> dscp-cos-mapping]
[dscp-cos-mapping]

Table 18.1:Syntax Descriptions

Arguments...	Description...
ipv6 access-list <acl name>	Enables the IPv6 configuration level and defines the name of the IPv6 ACL. The <acl name> can contain up to 199 characters and numbers, but cannot begin with a number and cannot contain any spaces or quotation marks.
permit	The ACL will permit (forward) packets that match a policy in the access list.
deny	The ACL will deny (drop) packets that match a policy in the access list.
icmp	Indicates the you are filtering ICMP packets.

Table 18.1:Syntax Descriptions

Arguments...	Description...
protocol	The type of IPv6 packet you are filtering. You can specify a well-known name for some protocols whose number is less than 255. For other protocols, you must enter the number. Enter “?” instead of a protocol to list the well-known names recognized by the CLI. IPv6 protocols include: • **AHP** – Authentication Header • **ESP** – Encapsulating Security Payload • **IPv6** – Internet Protocol version 6 • **SCTP** – Stream Control Transmission Protocol
<ipv6-source-prefix>/<prefix-length>	The <ipv6-source-prefix>/<prefix-length> parameter specify a source prefix and prefix length that a packet must match for the specified action (deny or permit) to occur. You must specify the <ipv6-source-prefix> parameter in hexadecimal using 16-bit values between colons as documented in RFC 2373. You must specify the <prefix-length> parameter as a decimal value. A slash mark (/) must follow the <ipv6-prefix> parameter and precede the <prefix-length> parameter.
<ipv6-destination-prefix>/<prefix-length>	The <ipv6-destination-prefix>/<prefix-length> parameter specify a destination prefix and prefix length that a packet must match for the specified action (deny or permit) to occur. You must specify the <ipv6-destination-prefix> parameter in hexadecimal using 16-bit values between colons as documented in RFC 2373. You must specify the <prefix-length> parameter as a decimal value. A slash mark (/) must follow the <ipv6-prefix> parameter and precede the <prefix-length> parameter
any	When specified instead of the <ipv6-source-prefix>/<prefix-length> or <ipv6-destination-prefix>/<prefix-length> parameters, matches any IPv6 prefix and is equivalent to the IPv6 prefix::/0.
host	Allows you specify a host IPv6 address. When you use this parameter, you do not need to specify the prefix length. A prefix length of all128 is implied.
icmp-type	ICMP packets can be filtered by ICMP message type. The type is a number from 0 to 255.
icmp code	ICMP packets, which are filtered by ICMP message type can also be filtered by the ICMP message code. The code is a number from 0 to 255,
icmp-message	ICMP packets are filtered by ICMP messages. See “ICMP Message Configurations” on page 18-10 for a list of ICMP message types.
tcp	Indicates the you are filtering TCP packets.
udp	Indicates the you are filtering UDP packets.

Table 18.1:Syntax Descriptions

Arguments...	Description...
<ipv6-source-prefix>/<prefix-length>	The <ipv6-source-prefix>/<prefix-length> parameter specify a source prefix and prefix length that a packet must match for the specified action (deny or permit) to occur. You must specify the <ipv6-source-prefix> parameter in hexadecimal using 16-bit values between colons as documented in RFC 2373. You must specify the <prefix-length> parameter as a decimal value. A slash mark (/) must follow the <ipv6-prefix> parameter and precede the <prefix-length> parameter.
<ipv6-destination-prefix>/<prefix-length>	The <ipv6-destination-prefix>/<prefix-length> parameter specify a destination prefix and prefix length that a packet must match for the specified action (deny or permit) to occur. You must specify the <ipv6-destination-prefix> parameter in hexadecimal using 16-bit values between colons as documented in RFC 2373. You must specify the <prefix-length> parameter as a decimal value. A slash mark (/) must follow the <ipv6-prefix> parameter and precede the <prefix-length> parameter
any	When specified instead of the <ipv6-source-prefix>/<prefix-length> or <ipv6-destination-prefix>/<prefix-length> parameters, matches any IPv6 prefix and is equivalent to the IPv6 prefix::/0.
host	Allows you specify a host IPv6 address. When you use this parameter, you do not need to specify the prefix length. A prefix length of all128 is implied.
tcp-udp-operator	The <tcp-udp-operator> parameter can be one of the following: • **eq** – The policy applies to the TCP or UDP port name or number you enter after **eq**. • **gt** – The policy applies to TCP or UDP port numbers greater than the port number or the numeric equivalent of the port name you enter after **gt**. Enter "?" to list the port names. • **lt** – The policy applies to TCP or UDP port numbers that are less than the port number or the numeric equivalent of the port name you enter after **lt**. • **neq** – The policy applies to all TCP or UDP port numbers except the port number or port name you enter after **neq**. • **range** – The policy applies to all TCP port numbers that are between the first TCP or UDP port name or number and the second one you enter following the **range** parameter. The range includes the port names or numbers you enter. For example, to apply the policy to all ports between and including 23 (Telnet) and 53 (DNS), enter the following: **range 23 53**. The first port number in the range must be lower than the last number in the range. The <source-port number> and <destination-port-number> for the tcp-udp-operator is the number of the port.

Table 18.1:Syntax Descriptions

Arguments...	Description...
ipv6-operator	Allows you to filter the packets further by using one of the following options: • **dscp** – The policy applies to packets that match the traffic class value in the traffic class field of the IPv6 packet header. This operator allows you to filter traffic based on TOS or IP precedence. You can specify a value from 0 – 63. • **fragments** – The policy applies to fragmented packets that contain a non-zero fragment offset. **NOTE:** This option is not applicable to filtering based on source or destination port, TCP flags, and ICMP flags. • **routing** – The policy applies only to IPv6 source-routed packets. **NOTE:** This option is not applicable to filtering based on source or destination port, TCP flags, and ICMP flags. •
802.1p-priority-matching <number>	If you want to match only those packets that have the same 802.1p priorities as specified in the ACL. Enter 0 – 7.
dscp-marking <number>	Use the **dscp-marking** <number> parameter to specify a new QoS value to the packet. If a packet matches the filters in the ACL statement, this parameter assigns the DSCP value that you specify to the packet. Enter 0 – 63.
802.1p-priority-marking <number>	Use the **802.1p-priority-marking** <number> parameter to specify a new QoS value to the packet. If a packet matches the filters in the ACL statement, this parameter assigns the 802.1p priority that you specify to the packet. Enter 0 – 7.
internal-priority-marking <number>	Use the **internal-priority-marking** <number> parameter to specify a new QoS value to the packet. If a packet matches the filters in the ACL statement, this parameter assigns the internal priority that you specify to the packet. Enter 0 – 7.
dscp-marking <number>	Use the **dscp-marking** <number> **dscp-cos-mapping** parameters parameters to specify a DSCP value and map that value to an internal QoS table to obtain the packet's new QoS value. The following occurs when you use these parameters. • You enter 0 – 63 for the **dscp-marking** <number> parameter. • The **dscp-cos-mapping** parameter takes the DSCP value you specified and compares it to an internal QoS table, which is indexed by DSCP values. The corresponding 802.1p priority, internal forwarding priority, and DSCP value is assigned to the packet.

Table 18.1:Syntax Descriptions

Arguments...	Description...
dscp-cos-mapping	Use **dscp-cos-mapping** if you want to use the DSCP value in the packet's header to alter its QoS value. When you enter **dscp-cos-mapping**, the DSCP value in the packet's header is compared to a column in the internal QoS table. The 802.1p priority, internal forwarding priority, and DSCP value that are mapped to the matching column is assigned to the packet.

ICMP Message Configurations

If you want to specify an ICMP message, you can enter one of the following:

- beyond-scope
- destination-unreachable
- echo-reply
- echo-request
- header
- hop-limit
- mld-query
- mld-reduction
- mld-report
- nd-na
- nd-ns
- next-header
- no-admin
- no-route
- packet-too-big
- parameter-option
- parameter-problem
- port-unreachable
- reassembly-timeout
- renum-command
- renum-result
- renum-seq-number
- router-advertisement
- router-renumbering
- router-solicitation
- sequence
- time-exceeded
- unreachable

NOTE: If you do not specify a message type, the ACL applies to all types ICMP messages types.

Applying an IPv6 ACL to an Interface

To apply an IPv6 ACL to an interface, enter commands such as the following:

```
FastIron(config)# interface ethernet 3/1
FastIron(config-if-e100-3/1)# ipv6 traffic-filter access1 in
```

This example applies the IPv6 ACL "access1" to incoming IPv6 packets on Ethernet interface 3/1. As a result, Ethernet interface 3/1 denies all incoming packets from the site-local prefix fec0:0:0:2::/64 and the global prefix 2001:100:1::/48 and permits all other incoming packets.

Syntax: ipv6 traffic-filter <ipv6-acl-name> in

For the <ipv6-acl-name> parameter, specify the name of an IPv6 ACL created using the **ipv6 access-list** command.

The **in** keyword applies the specified IPv6 ACL to incoming IPv6 packets on the interface.

Displaying ACLs

To display the ACLs configured on a device, enter the **show ipv6 access-list** command. Here is an example:

```
FastIron# show ipv6 access-list
ipv6 access-list v6-acl1: 1 entries
 deny ipv6 any any
ipv6 access-list v6-acl2: 1 entries
 permit ipv6 any any
ipv6 access-list v6-acl3: 2 entries
 deny ipv6 2001:aa:10::/64 any
 permit ipv6 any any
ipv6 access-list v6-acl4: 2 entries
 deny ipv6 2002:aa::/64 any
 permit ipv6 any any
ipv6 access-list rate-acl: 1 entries
 permit ipv6 any any traffic-policy rate800M
ipv6 access-list v6-acl5: 8 entries
 permit tcp 2002:bb::/64 any
 permit ipv6 2002:bb::/64 any
 permit ipv6 2001:aa:101::/64 any
 permit ipv6 2001:aa:10::/64 2001:aa:102::/64
 permit ipv6 host 2001:aa:10::102 host 2001:aa:101::102
 permit ipv6 host 2001:aa:10::101 host 2001:aa:101::101 dscp-matching 0 dscp-
marking 63 dscp-cos-mapping
 permit ipv6 any any dscp-matching 63 dscp-cos-mapping
 permit ipv6 any any fragments
```

Syntax: show ipv6 access-list [<access-list-name>]

Chapter 19

Configuring Port Mirroring and Monitoring

The procedures in this chapter describe how to configure port mirroring for FastIron devices.

Port mirroring is a method of monitoring network traffic that forwards a copy of each incoming or outgoing packet from one port on a network switch to another port where the packet can be analyzed. Port mirroring may be used as a diagnostic tool or debugging feature, especially for preventing attacks. Port mirroring can be managed locally or remotely.

Configure port mirroring by assigning a port from which to copy all packets, and a “mirror” port where the copies of the packets are sent (also known as the monitor port). A packet received on, or issued from, the first port is forwarded to the second port as well. Attach a protocol analyzer on the mirror port to monitor each segment separately. The analyzer captures and evaluates the data without affecting the client on the original port.

The mirror port may be a port on the same switch with an attached RMON probe, a port on a different switch in the same hub, or the switch processor.

NOTE: Port mirroring can consume significant CPU resources while active.

Mirroring Support across FastIron Platforms

Table 19.1lists which FastIron devices support the mirroring features discussed in this chapter.

Table 19.1: Mirroring Support for FastIron Devices

Mirroring Feature	FESX	FSX	FGS and FLS
Basic Port Mirroring and Monitoring	X	X	X
ACL-Based Inbound Mirroring	X	X	X
MAC Filter-Based Mirroring			X
VLAN-Based Mirroring			X

Configuring Port Mirroring and Monitoring

FastIron devices support monitoring of both inbound and outbound traffic on individual ports. To configure port monitoring, specify the ***mirror port***, then enable monitoring on the ***monitored port***.

- The ***mirror port*** is the port to which the monitored traffic is copied. Attach your protocol analyzer to the mirror port.
- The ***monitored port*** is the port whose traffic you want to monitor.

Configuration Notes

Refer to the following rules when configuring port mirroring and monitoring:

- Port monitoring and sFlow support:
  - FWSX devices and FESX/FSX devices running software release 02.2.01 or later, support sFlow and inbound port monitoring together on the same device, however, these devices *do not* support port monitoring and sFlow together within the same port region. See the section “About Port Regions” on page 8-1 for a list of valid port ranges on these devices.
  - FastIron GS and LS devices support sFlow and port monitoring together on the same port.
- Table 19.2 lists the number of mirror and monitor ports supported on the FastIron devices.

Table 19.2: Number of Mirror and Monitored Ports Supported

Port type	**Maximum number supported on...**	
	FastIron GS and LS	**FastIron X Series**
Ingress mirror ports	1	1 per port region[1]
Egress mirror ports	1	1 per port region[1]
Ingress monitored ports	no limit	no limit
Egress monitored ports	no limit	8

1.FastIron X Series devices support multiple ingress and egress mirror ports. For 1-Gigabit ports, ports in groups of 12 share one ingress mirror port and one egress mirror port. So ports 1 and 2 cannot have different mirror ports, but ports 1 and 13 can. Each 10-Gigabit port can have one ingress mirror port and one egress mirror port.

- You can configure a mirror port specifically as an ingress port, an egress port, or both.
- Mirror ports can run at any speed and are not related to the speed of the ingress or egress monitored ports.
- The same port cannot be both a monitored port and the mirror port.
- The same port can be monitored by one mirror port for ingress traffic and another mirror port for egress traffic.
- The mirror port cannot be a trunk port.
- The monitored port and its mirror port do not need to belong to the same port-based VLAN.
  - If the mirror port is in a *different* VLAN from the monitored port, the packets are tagged with the monitor port’s VLAN ID.
  - If the mirror port is in the *same* VLAN as the monitored port, the packets are tagged or untagged, depending on the mirror port’s configuration.

NOTE: The FastIron GS and LS perform "as-is" mirroring. Mirrored packets are sent out of the mirror port in the VLAN tagged or untagged format in which they are received or sent on the monitor port. The VLAN tagged or untagged format is not changed.

- More than one monitored port can be assigned to the same mirror port.
- If the primary interface of a trunk is enabled for monitoring, the entire trunk will be monitored. You can also enable an individual trunk port for monitoring using the **config-trunk-ind** command.

Command Syntax

To configure port monitoring, enter commands such as the following:

```
FastIron(config)#mirror-port ethernet 4
FastIron(config)#interface ethernet 11
FastIron(config-if-e1000-11)#monitor ethernet 4 both|in|out
```

Syntax: [no] mirror-port ethernet [<stacknum>/<slotnum>/]<portnum> [input | output]

Syntax: [no] monitor ethernet [<stacknum>/<slotnum>/]<portnum> both | in | out

The <stacknum> parameter is required on FastIron GS devices running software release 02.5.00 or later.

The <slotnum> parameter is required on chassis devices and on FastIron GS devices running software release 02.5.00 or later.

The <portnum> parameter specifies the mirror port to which the monitored port's traffic will be copied.

The [input | output] parameters configure the mirror port exclusively for ingress or egress traffic. If you do not specify one, both types of traffic apply.

The **both | in | out** parameters specify the traffic direction you want to monitor on the mirror port. There is no default.

To display the port monitoring configuration, enter the **show monitor** and **show mirror** commands.

NOTE: Port regions do not apply to trunk group configurations on the X-Series devices. However, port regions do apply to port monitoring and unknown unicast configurations.

FastIron Edge Switch X424 and X424HF, and FastIron Workgroup Switch X424:

- Ports 1 – 12
- Ports 13 – 24
- Port 25
- Port 26

FastIron Edge Switch X448 and FastIron Workgroup Switch X448:

- Ports 1 – 12
- Ports 13 – 24
- Port 25 – 36
- Port 37 – 48
- Port 49
- Port 50

FastIron SuperX:

- Management Module:
    - Ports 1 – 12

- 24-port Gigabit Ethernet Copper Interface Module
    - Ports 1 – 12
    - Ports 13 – 24
- 24-port Gigabit Ethernet Fiber Interface Module:
    - Ports 1 – 12
    - Ports 13 – 24
- 2-port 10-Gigabit Ethernet Fiber Interface Module
    - Port 1
    - Port 2

ACL-Based Inbound Mirroring

Platform Support:

- FESX/FSX/FWSX devices running software release 03.2.00 and later
- FGS and FLS devices running software release 04.0.00 and later

FastIron devices can use an ACL to specify a stream of traffic monitor and mirror it to a specified physical port.

The procedure for configuring ACL-Based mirroring consists of the following steps:

- Define a mirror port as described in “Specifying the Destination Mirror Port” on page 19-4
- Create an ACL with a mirror clause
- Apply the ACL to an interface
- Specify the Destination mirror port

Creating an ACL with Mirror Clause

The **mirror** keyword has been added for inclusion in ACL clauses to direct traffic that meets the clause to be sent to mirror another port. In the following example, the ACL is used to direct IP traffic to a mirror port:

```
FastIron(config)#access-list 101 permit ip any any mirror
```

The **mirror** parameter directs selected traffic to the mirrored port. Traffic to be mirrored can only be selected using the **permit** clause and is only supported on Layer-3 ACLs. Deny traffic is dropped and not mirrored.

Specifying the Destination Mirror Port

You can specify physical ports or a trunk to mirror traffic from. If you complete the rest of the configuration but do not specify a destination mirror port, the port-mirroring ACL will be non-operational. This can be useful if you want to be able to mirror traffic by a set criteria on-demand. With this configuration, you just configure a destination mirror port whenever you want the port-mirroring ACL to become operational.

The following sections describe how to specify a destination port for a port or a trunk as well as the special considerations required when mirroring traffic from a virtual interface.

Specifying the Destination Mirror Port for Physical Ports

When you want traffic that has been selected by ACL-based Inbound Mirroring to be mirrored, you must configure a destination mirror port. This configuration is performed at the Interface Configuration of the port whose traffic you are mirroring. The destination port must be the same for all ports in a port region as described in “Ports from a Port Region must be Mirrored to the Same Destination Mirror port” on page 19-5.

In the following example, ACL mirroring traffic from port 1/1 is mirrored to port 1/3:

```
FastIron(config)#interface ethernet 1/1
FastIron(config-if-e10000-1/1)#acl-mirror-port ethernet 1/3
```

Syntax: [no] acl-mirror-port ethernet [<stacknum>]/[<slotnum>]|<portnum>

The [<stacknum>]/[<slotnum>]|<portnum> variable specifies port to which ACL-mirror traffic from the configured interface will be mirrored.

The <stacknum> parameter is required on FastIron GS devices running software release 02.5.00 or later.

The <slotnum> parameter is required on chassis devices and on FastIron GS devices running software release 02.5.00 or later.

The <portnum> parameter specifies the mirror port to which the monitored port's traffic will be copied.

Ports from a Port Region must be Mirrored to the Same Destination Mirror port

Port regions as described in "About Port Regions" on page 8-1 are important when defining a destination mirror port. This is because all traffic mirrored from any single port in a port region will be mirrored to the same destination mirror port as traffic mirrored from any other port in the same port region. For example, ports 1/1 to 1/12 are in the same port region. If you configure ports 1/1 and 1/2 to mirror their traffic, they should use the same destination mirror port as shown in the following configuration:

```
FastIron(config)#interface ethernet 1/1
FastIron(config-if-e10000-1/1)#acl-mirror-port ethernet 2/3
FastIron(config)#interface ethernet 1/2
FastIron(config-if-e10000-1/2)#acl-mirror-port ethernet 2/3
```

If ports within the same port region are mirrored to different destination ports, an error message will be generated as shown in the following example, and the configuration will be disallowed:

```
FastIron(config)#interface ethernet 1/1
FastIron(config-if-e10000-1/1)#acl-mirror-port ethernet 4/3
FastIron(config)#interface ethernet 1/2
FastIron(config-if-e10000-1/2)#acl-mirror-port ethernet 4/7
Error - Inbound Mirror port 4/3 already configured for port region 1/1 - 1/12
```

When a destination port is configured for any port within a port region, traffic from any ACL with a mirroring clause assigned to any port in that port region will be mirrored to that destination port. This will occur even if a destination port is not explicitly configured for the port with the ACL configured. In the following example, an ACL with a mirroring clause (101) is applied to a port (1/1). Another port in the same region (1/3) has a destination port set (4/3). In this example, traffic generated from operation of ACL 101 is mirrored to port 4/3 even though a destination port has not explicitly been defined for traffic from port 1/1.

```
FastIron(config)#interface ethernet 1/1
FastIron(config-if-e10000-1/1)#ip access-group 101 in
FastIron(config)#interface ethernet 1/3
FastIron(config-if-e10000-1/3)#acl-mirror-port ethernet 4/3
```

NOTE: If a destination mirror port is not configured for any ports within the port region where the port-mirroring ACL is configured, the ACL will not mirror the traffic but the ACL will be applied to traffic on the port.

Specifying the Destination Mirror Port for Trunk Ports

You can mirror the traffic that has been selected by ACL-based Inbound Mirroring from a trunk by configuring a destination port for the primary port within the trunk configuration as shown:

```
FastIron(config)#trunk ethernet 1/1 to 1/4
FastIron(config)#interface ethernet 1/1
FastIron(config-if-e10000-1/1)#acl-mirror-port ethernet 1/8
```

Using this configuration, all trunk traffic will be mirrored to port 1/8.

Limitations When Configuring ACL-based Mirroring with Trunks

The **config-trunk-ind** option as described in "Disabling or Re-Enabling a Trunk Port" on page 13-12 cannot operate with ACL-based mirroring as described in the following:

- If a trunk is configured with the **config-trunk-ind** option, ACL-based mirroring will not be allowed.

- If the **config-trunk-ind** option is added to a trunk, any ports that are configured for ACL-based Mirroring will have monitoring removed and the following message will be displayed:

  Trunk port monitoring, if any, has been removed.

If an individual port is configured for ACL-based Mirroring, you cannot add it to a trunk.

- If you try to add a port that is configured for ACL-based Mirroring to a trunk, the following message appears:

  Note - acl-mirror-port configuration is removed from port 2 in new trunk.

NOTE: If you want to add a port configured for ACL-based Mirroring to a trunk, you must first remove the **acl-mirror-port** from the port configuration. You can then add the port to a trunk that can then be configured for ACL-based trunk mirroring.

Behavior of ACL-based Mirroring when Deleting trunks

If you delete a trunk that has ACL-based mirroring configured, the ACL-based mirroring configuration will be configured on the individual ports that made up the trunk.

For example, if a trunk is configured as shown in the following example and is then deleted from the configuration as shown, each of the ports that previously were contained in the trunk will be configured for ACL-based mirroring.

```
FastIron(config)#trunk ethernet 4/1 to 4/2
FastIron(config)#trunk deploy
FastIron(config)#interface ethernet 4/1
FastIron(config-if-e10000)#acl-mirror-port ethernet 5/3
```

Deleting the trunk:

```
FastIron(config)#no trunk ethernet 4/1 to 4/2
```

Configuration for ACL-based mirroring on ports 4/1 and 4/2 that results from the trunk being deleted:

```
interface ethernet 4/1
   acl-mirror-port ethernet 5/3

interface ethernet 4/2
   acl-mirror-port ethernet 5/3
```

Configuring ACL-based Mirroring for ACLs bound To Virtual Interfaces

For configurations that have an ACL configured for ACL-based mirroring bound to a virtual interface, you must configure the **acl-mirror-port** command on a physical port that is a member of the same VLAN as the virtual interface. Additionally, only traffic that arrives at ports that belong to the same port group as the physical port where the **acl-mirror-port** command is configured will be mirrored. This follows the same rules described in “Ports from a Port Region must be Mirrored to the Same Destination Mirror port” on page 19-5.

For example, in the following configuration ports 4/1, 4/2 and 5/2 are in VLAN 10 with ve 10. Ports 4/1 and 4/2 belong to the same port group while port 5/3 belongs to another port group:

```
FastIron(config)#vlan 10
FastIron(config-vlan-10)#tagged ethernet 4/1 to 4/2
FastIron(config-vlan-10)#tagged ethernet 5/3
FastIron(config-vlan-10)#router-interface ve 10

FastIron(config)#interface ethernet 4/1
FastIron(config-if-e10000-4/1)#acl-mirror-port ethernet 5/1

FastIron(config)#interface ve 10
FastIron(config-vif-10)#ip address 10.10.10.254/24
FastIron(config-vif-10)#ip access-group 102 in

FastIron(config)#access-list 102 permit ip any any mirror
```

In this configuration, the **acl-mirror-port** command is configured on port 4/1 which is a member of ve 10. Because of this, ACL-based mirroring will only apply to VLAN 10 traffic that arrives on ports 4/1 and 4/2. It will not apply to VLAN 10 traffic that arrives on port 5/3 because that port belongs to a different port group than ports 4/1 and 4/2. This is because if you apply ACL-based mirroring on an entire VE, and enable mirroring in only one port region, traffic that is in the same VE but on a port in a different port region will not be mirrored.

To make the configuration apply ACL-based mirroring to VLAN 10 traffic arriving on port 5/3, you must add the following command to the configuration:

```
FastIron(config)#interface ethernet 5/3
FastIron(config-if-e10000-5/3)#acl-mirror-port ethernet 5/1
```

If a port is in both mirrored and non-mirrored VLANs, only traffic on the port from the mirrored VLAN will be mirrored. For example, the following configuration adds VLAN 20 to the previous configuration. In this example, ports 4/1 and 4/2 are in both VLAN 10 and VLAN 20. ACL-based mirroring is only applied to VLAN 10. Consequently, traffic that is on ports 4/1 and 4/2 that belongs to VLAN 20 will not be mirrored.

```
FastIron(config)#vlan 10
FastIron(config-vlan-10)#tagged ethernet 4/1 to 4/2
FastIron(config-vlan-10)#tagged ethernet 5/3
FastIron(config-vlan-10)#router-interface ve 10

FastIron(config)#vlan 20
FastIron(config-vlan-20)#tagged ethernet 4/1 to 4/2

FastIron(config)#interface ethernet 4/1
FastIron(config-if-e10000-4/1)#acl-mirror-port ethernet 5/1

FastIron(config)#interface ve 10
FastIron(config-vif-10)#ip address 10.10.10.254/24
FastIron(config-vif-10)#ip access-group 102 in

FastIron(config)#access-list 102 permit ip any any mirror
```

MAC Filter-Based Mirroring

Platform Support:

- FGS and FLS devices running software release 04.0.00 and later

This feature allows traffic entering an ingress port to be monitored from a mirror port connected to a data analyzer, based on specific source and destination MAC addresses. This feature supports mirroring of inbound traffic only. Outbound mirroring is not supported.

MAC-Filter-Based Mirroring allows a user to specify a particular stream of data for mirroring using a filter, eliminating the need to analyze all incoming data to the monitored port. To configure MAC-Filter-Based Mirroring, the user must perform three steps:

- Define a mirror port
- Create a MAC filter with a mirroring clause
- Apply the MAC filter to an interface

The following sections describe these steps.

Define a Mirror Port

To activate mirroring on a port, use the mirror command in the global configuration mode. For example:

```
FastIron(config)#mirror e 0/1/14
```

NOTE: If there is no input mirror port configured, MAC-Filter Based Mirroring does not take effect. It remains in the configuration, but is not activated.

FGS and FLS devices support one ingress mirror and one egress mirror per system. These ports are shared by all mirroring features - port-based mirroring, VLAN-based mirroring, ACL-based mirroring and MAC-based mirroring.

NOTE: Port-based mirroring, VLAN mirroring, and MAC-filter based mirroring can be enabled on a port at the same time. In this case, the preference order is Port, VLAN, and MAC-filter.

Create a MAC Filter with a Mirroring Clause

The keyword "mirror" is added to MAC filter clauses to direct desired traffic to the mirror port. In the following examples, the MAC filter directs traffic to a mirror port:

```
FastIron(config)#mac filter 1 permit 0000.1111.2222.ffff.ffff.ffff
0000.2222.3333.ffff.ffff.fff mirror
```

In this example, any flow matching the SA (source address) 0000.1111.2222 and the DA (destination address) 0000.2222.3333 will be mirrored. Other flows will not be mirrored.

Apply the MAC Filter to an Interface

Apply the MAC filter to an interface using the **mac-filter-group** command, as shown:

```
FastIron(config)#interface ethernet 0/1/1
FastIron(config-if-e10000-0/1/1)#mac filter-group 1
```

VLAN-Based Mirroring

Platform Support:

- FGS and FLS devices running software release 03.0.00 and later

The VLAN-Based MIrroring feature allows users to monitor all incoming traffic in one or more VLANs by sending a mirror image of that traffic to a port that is configured as the mirror port. This feature meets the requirements of CALEA (Communications Assistance for Law Enforcement Act of 1994).

CLI Command

Configure this feature using the monitor command, in VLAN configuration mode:

```
FastIron(config-VLAN-20)#monitor
```

Syntax: [no] monitor

EXAMPLES:

To enable mirroring on VLANs 10 and 20, to mirror port e 0/1/21, enter the following commands:

```
FastIron(config)#mirror-port ethernet 0/1/21 input
System-wide ingress mirror port set to port 0/1/21
FastIron(config)#vlan 10
FastIron(config-VLAN-10)#monitor
FastIron(config)#vlan 20
FastIron(config-VLAN-20)#monitor
FastIron(config-VLAN-20)#end
```

To disable mirroring on VLAN 20, enter the following commands:

```
FastIron(config)#vlan 20
FastIron(config-VLAN-20)#no monitor
FastIron(config-VLAN-20)#end
```

Displaying VLAN Mirroring Status

The **show vlan** command displays the vlan mirroring status:

```
FastIron#show vlans
Total PORT-VLAN entries: 4
Maximum PORT-VLAN entries: 4060

Legend: [Stk=Stack-Id, S=Slot]

PORT-VLAN 1, Name DEFAULT-VLAN, Priority level0, Spanning tree On
 Untagged Ports: (Stk0/S1)   3   4   5   6   7   8   9  10  11  12  13  14
 Untagged Ports: (Stk0/S1)  15  16  17  18  19  20  21  22  23  24  25  26
 Untagged Ports: (Stk0/S1)  27  28  29  30  31  32  33  34  35  36  37  38
 Untagged Ports: (Stk0/S1)  39  40  41  42  43  44  45  46  47  48
 Untagged Ports: (Stk0/S2)   1   2
   Tagged Ports: None
   Uplink Ports: None
 DualMode Ports: None
 Mac-Vlan Ports: None
     Monitoring: Disabled
PORT-VLAN 10, Name [None], Priority level0, Spanning tree On
 Untagged Ports: (Stk0/S1)   1
   Tagged Ports: None
   Uplink Ports: None
 DualMode Ports: None
 Mac-Vlan Ports: None
     Monitoring: Enabled
PORT-VLAN 20, Name [None], Priority level0, Spanning tree On
 Untagged Ports: (Stk0/S1)   2
   Tagged Ports: None
   Uplink Ports: None
 DualMode Ports: None
 Mac-Vlan Ports: None
     Monitoring: Disabled
```

Configuration Notes

The following statements apply to VLAN-Based Mirroring configurations:

1. A VLAN must have at least one port member configured before “monitor” can be configured.
2. To activate mirroring, configure the input mirror port using the **mirror** command in global configuration mode.
3. Multiple VLANs can have “monitor” enabled at the same time, and there is no limit to the number of monitor-configured VLANs.
4. FGS and FLS switches support one ingress and one egress mirror per system. These mirror ports are shared by all mirroring features; port-based mirroring, VLAN-based mirroring, ACL-based mirroring, and MAC-filter-based mirroring.
5. The mirror port is subject to the same scheduling and bandwidth management as the other ports in the system. If the amount of traffic being sent to the mirror port exceeds the available bandwidth, some of that traffic may be dropped.
6. All incoming traffic (tagged and untagged) in the VLAN is mirrored. Mirroring is “as-is”, and is not affected by the configuration of the mirror port itself. Incoming tagged traffic is set out tagged and incoming untagged traffic is sent out untagged, regardless of which VLANs the mirror port belongs to, and whether the mirror port is tagged or untagged.

This feature is supported on Layer 2 and Layer 3 images.

Chapter 20

Configuring Base Layer 3 and Enabling Routing Protocols

The Layer 2 with Base Layer 3 software image contains all the system-level features in the Layer 2 images, along with the following:

- Static IP routes
- RIPv1 and RIPv2 (see note, below)
- Routing between directly connected subnets
- RIP advertisements of the directly connected subnets
- Virtual interfaces
- VRRP

NOTE:

- Layer 2 with Base Layer 3 images provide static RIP support. The device does not learn RIP routes from other Layer 3 devices. However, the device does advertise directly connected routes and can be configured to dynamically learn default routes. Foundry Networks recommends that you deploy these devices only at the edge of your network, since incoming traffic can learn directly-connected routes advertised by the Foundry device, but outgoing traffic to other devices must use statically configured or default routes.
- The Base Layer 3 images do not support IP multicasting, OSPF, or BGP4.
- Base Layer 3 feature support has been added to FGS and FLS devices effective with release 04.0.00.
- The FWSX devices are Layer 2 switches only. They do not support Base Layer 3 and full Layer 3 features.

Adding a Static IP Route

Platform Support:

- FESX/FSX/FWSX devices – all software releases
- FGS/FLS devices running software release 04.0.00 and later

To add a static IP route, enter a command such as the following at the global CONFIG level of the CLI:

```
FastIron(config)#ip route 209.157.2.0 255.255.255.0 192.168.2.1
```

This command adds a static IP route to the 209.157.2.x/24 subnet.

Syntax: [no] ip route <dest-ip-addr> <dest-mask> <next-hop-ip-addr> [<metric>]

or

Syntax: [no] ip route <dest-ip-addr>/<mask-bits> <next-hop-ip-addr> [<metric>]

On FastIron X Series, FastIron GS, and FastIron LS devices, you can add up to 1024 static IP routes. However, on FastIron X Series devices running software release 03.2.00, up to 2048 static IP routes can be configured.

The <dest-ip-addr> is the route's destination. The <dest-mask> is the network mask for the route's destination IP address. Alternatively, you can specify the network mask information by entering a forward slash followed by the number of bits in the network mask. For example, you can enter 192.0.0.0 255.255.255.0 as 192.0.0.0/.24. To configure a default route, enter 0.0.0.0 for <dest-ip-addr> and 0.0.0.0 for <dest-mask> (or 0 for the <mask-bits> if you specify the address in CIDR format). Specify the IP address of the default gateway using the <next-hop-ip-addr> parameter.

The <next-hop-ip-addr> is the IP address of the next-hop router (gateway) for the route.

The <metric> parameter specifies the cost of the route and can be a number from 1 – 16. The default is 1. The metric is used by RIP. If you do not enable RIP, the metric is not used.

NOTE: You cannot specify **null0** or another interface as the next hop in the Base Layer 3 image.

Adding a Static ARP Entry

Static entries are useful in cases where you want to pre-configure an entry for a device that is not connected to the Foundry device, or you want to prevent a particular entry from aging out. The software removes a dynamic entry from the ARP cache if the ARP aging interval expires before the entry is refreshed. Static entries do not age out, regardless of whether the Foundry device receives an ARP request from the device that has the entry's address. The software places a static ARP entry into the ARP cache as soon as you create the entry.

To add a static ARP entry, enter a command such as the following at the global CONFIG level of the CLI:

```
FastIron(config)#arp 1 209.157.22.3 aaaa.bbbb.cccc ethernet 3
```

This command adds a static ARP entry that maps IP address 209.157.22.3 to MAC address aaaa.bbbb.cccc. The entry is for a MAC address connected to FastIron port 3.

Syntax: [no] arp <num> <ip-addr> <mac-addr> ethernet [<slotnum>/]<portnum>

The <num> parameter specifies the entry number. You can specify a number from 1 up to the maximum number of static entries allowed on the device. You can allocate more memory to increase this amount. To do so, enter the **system-max ip-static-arp** <num> command at the global CONFIG level of the CLI.

The <ip-addr> command specifies the IP address of the device that has the MAC address of the entry.

The <mac-addr> parameter specifies the MAC address of the entry.

The <portnum> command specifies the port number attached to the device that has the MAC address of the entry. If you are configuring a chassis device, specify the slot number as well as the port number (<slotnum>/<portnum>).

NOTE: The **clear arp** command clears learned ARP entries but does not remove any static ARP entries.

Modifying and Displaying Layer 3 System Parameter Limits

This section shows how to view and configure some of the Layer 3 system parameter limits.

Configuration Notes

- Changing the system parameters reconfigures the device's memory. Whenever you reconfigure the memory on a Foundry device, you must save the change to the startup-config file, then reload the software to place the change into effect.

- The Layer 3 system parameter limits for FastIron IPv6 models are automatically adjusted by the system and cannot be manually modified. See “FastIron IPv6 Models” on page 20-5.

FGS with Base Layer 3

NOTE: This section applies only to FGS Base-L3 build. It does not apply to other platforms such as FESX or SuperX.

FGS base-L3 uses TCAMs for IP routing, access list and MAC based vlan and some other features. There is a total of 1024 TCAMs in an FGS.

You can set the number of TCAMs reserved for IP routing using the **system-max** command.

```
FastIron(config)#system-max hw-ip-route-tcam ??
```

The range is from 64 to 1020 and the default is 256.

The **show ip route** command displays the usage of TCAMs in routing:

EXAMPLES:

```
FastIron#show ip route
Total number of IP routes: 29
Route TCAM: total 64, used 64, add HW route failure=34

Start index: 1  D:Connected  R:RIP  S:Static  O:OSPF *:Candidate default
        Destination     NetMask         Gateway         Port       Cost   Type
1       2.2.10.0        255.255.255.0   0.0.0.0         v10        1      D
2       2.2.11.0        255.255.255.0   0.0.0.0         v11        1      D
...
```

In this example "add HW route failure" shows the number of failures in installing routing TCAM.

FGS Base L3 installs routing TCAMs for the interface subnet and static routes. A packet is routed in hardware if its destination IP address matches the interface or stack route subnet, and the nexthop or destination ARP is resolved. The unknown unicast packets have destination IP addresses that do not match any static route or interface subnets.

The handling of unknown unicast packets differs depending on TCAM availability. For example, for a system that never encounters out-of-TCAM situation, If the default route is not configured or its next hop ARP is not resolved, all unknown unicast packets are handled based on the default L2 behavior. They are either dropped in hardware if "route-only" is configured, or VLAN-flooded in hardware.

NOTE: FGS Base-L3 does not allow "route-only" config.

If the default route is configured and its next hop ARP is resolved, unknown unicast packets are hardware-routed to the next hop, and not VLAN-flooded.

Once the FGS runs out of TCAM, it traps the unknown unicast packets to CPU for processing. If the default route is defined and its next hop ARP is resolved, the packets are routed by CPU. Otherwise, they follow the default L2 behavior. Because FGS Base L3 does not allow "route-only" configurations, these packets are VLAN- flooded.

The system does not change this CPU handling behavior back to hardware switching even when TCAMs again become available.

FastIron IPv4 Models

You can configure the following Layer 3 system parameters on FastIron IPv4 models:

- Number of IP next hops and IP route entries
- Number of hardware logical interfaces (physical port and VLAN pairs)
- Number of multicast output interfaces (clients)

These parameters are automatically enabled with pre-defined default values. You can however, adjust these values to conform with your network's topology.

To display the current settings for the Layer 3 system parameters, use the **show default value** command. See "Displaying Layer 3 System Parameter Limits" on page 20-5.

To modify the default settings for the Layer 3 system parameters, use the **system max** command at the Global CONFIG level of the CLI. See "Modifying Layer 3 System Parameter Limits on IPv4 Models" on page 20-4.

Modifying Layer 3 System Parameter Limits on IPv4 Models

The Layer 3 system parameter limits share the same hardware memory space and, by default, consume all of the hardware memory allocated for these Layer 3 limits. Therefore, to increase the limit for one of the parameters, you must first decrease one or both of the other parameters' limits. If you enter a value that exceeds the memory limit, the CLI will display an error message and the configuration will not take effect.

For example, if the network topology has a smaller number of IP next hops and routes, but has numerous multicast output interfaces, you could decrease the number of IP next hops and routes, then increase the number of multicast output interfaces. To do so, enter commands such as the following:

```
FastIron(config)#system-max hw-ip-next-hop 1024
FastIron(config)#system-max hw-ip-mcast-mll 2048
FastIron(config)#write mem
FastIron(config)#reload
```

Likewise, if the network topology does not have a large number of VLANs, and the VLANs configured on physical ports are not widely distributed, you could decrease the number of hardware logical interfaces, then increase the number of IP next hops and multicast output interfaces. To do so, enter commands such as the following:

```
FastIron(config)#system-max hw-logical-interface 2048
FastIron(config)#system-max hw-ip-next-hop 3072
FastIron(config)#system-max hw-ip-mcast-mll 2048
FastIron(config)#write mem
FastIron(config)#reload
```

Syntax: system max hw-ip-next-hop <num>

Syntax: system max hw-logical-interface <num>

Syntax: system max hw-ip-mcast-mll <num>

NOTE: The above commands are not supported on IPv6 devices. See "FastIron IPv6 Models" on page 20-5.

The **hw-ip-next-hop** <num> parameter specifies the maximum number of IP next hops and routes supported on the device. Note that the maximum number includes unicast next hops and multicast route entries. Enter a number from 100 to 6144. The default is 2048.

The **hw-logical-interface** <num> parameter specifies the number of hardware logical interface pairs (physical port and VLAN pairs) supported on the device. Enter a number from 0 to 4096. When this parameter is set to 4096 (the maximum), the limit is not enforced. If you enter a number less than 4096, the limit is the total number of physical port and VLAN pairs that are IP-enabled in the system. The default is 4096.

The **hw-ip-mcast-mll** <num> parameter specifies the maximum number of multicast output interfaces (clients) supported on the device. If a given source or group has clients in *n* tagged VLANs on the router, then *n* + 1 mll entries are consumed for that source or group entry. Enter a number from 0 to 4096. The default is 1024.

FastIron IPv6 Models

FastIron IPv6 models support the same Layer 3 system parameters that use hardware memory, as do FastIron IPv4 models. However, there are some configuration differences for IPv6 models versus IPv4 models. The differences are as follows:

- Number of IP next hops and IP route entries – 6144 maximum and default value. The system automatically calculates this value, based on the maximum number of VLANs supported system-wide.
- Number of hardware logical interfaces (physical port and VLAN pairs) – This value is the same as the maximum number of VLANs supported system-wide, so it is not configurable nor displayed in the **show default values** output in IPv6 models.
- Number of multicast output interfaces (clients) – 3072 maximum. This value is fixed in IPv6 models and cannot be modified. This system parameter occupies its own hardware memory space.

To display the current settings for the Layer 3 system parameters, use the **show default value** command. See “Displaying Layer 3 System Parameter Limits” .

Displaying Layer 3 System Parameter Limits

To display the Layer 3 system parameter defaults, maximum values, and current values, enter the **show default value** command at any level of the CLI.

The following shows an example output on a FastIron X Series IPV4 device.

```
FastIron#show default value

sys log buffers:50          mac age time:300 sec       telnet sessions:5

ip arp age:10 min           bootp relay max hops:4     ip ttl:64 hops
ip addr per intf:24

igmp group memb.:140 sec    igmp query:60 sec

ospf dead:40 sec            ospf hello:10 sec          ospf retrans:5 sec
ospf transit delay:1 sec

System Parameters     Default     Maximum     Current
ip-arp                4000        64000       4000
ip-static-arp         512         1024        512

some lines omitted for brevity....


hw-ip-next-hop        2048        6144        2048
hw-logical-interface  4096        4096        4096
hw-ip-mcast-mll       1024        4096        1024
```

The following shows an example output on a FastIron X Series IPV6 device.

```
FastIron#show default value

sys log buffers:50          mac age time:300 sec       telnet sessions:5

ip arp age:10 min           bootp relay max hops:4     ip ttl:64 hops
ip addr per intf:24

igmp group memb.:140 sec    igmp query:60 sec

ospf dead:40 sec            ospf hello:10 sec          ospf retrans:5 sec
ospf transit delay:1 sec

System Parameters     Default    Maximum    Current
ip-arp                4000       64000      4000
ip-static-arp         512        1024       512

some lines omitted for brevity....


hw-ip-next-hop        6144       6144       6144
hw-ip-mcast-mll       1024       4096       1024
hw-traffic-condition    50       1024         50
```

Configuring RIP

Platform Support:

- FESX/FSX/FWSX devices running software release 02.0.00 and later

RIP is disabled by default. If you want the Foundry device to use RIP, you must enable the protocol globally, then enable RIP on individual ports. When you enable RIP on a port, you also must specify the version (version 1 only, version 2 only, or version 1 compatible with version 2).

Optionally, you also can set or change the following parameters:

- Route redistribution – You can enable the software to redistribute static routes from the IP route table into RIP. Redistribution is disabled by default.
- Learning of default routes – The default is disabled.
- Loop prevention (split horizon or poison reverse) – The default is poison reverse.

Enabling RIP

RIP is disabled by default. To enable it, use the following CLI method. You must enable the protocol both globally and on the ports on which you want to use RIP.

To enable RIP globally, enter the following command:

```
FastIron(config)#router rip
```

Syntax: [no] router rip

To enable RIP on a port and specify the RIP version, enter commands such as the following:

```
FastIron(config-rip-router)#interface ethernet 1
FastIron(config-if-e1000-1)#ip rip v1-only
```

This command changes the CLI to the configuration level for port 1and enables RIP version 1 on the interface. You must specify the version.

Syntax: interface ethernet [<slotnum>/]<portnum>

Syntax: [no] ip rip v1-only | v1-compatible-v2 | v2-only

Enabling Redistribution of IP Static Routes into RIP

By default, the software does not redistribute the IP static routes in the route table into RIP. To configure redistribution, perform the following tasks:

- Configure redistribution filters (optional). You can configure filters to permit or deny redistribution for a route based on the route's metric. You also can configure a filter to change the metric. You can configure up to 64 redistribution filters. The software uses the filters in ascending numerical order and immediately takes the action specified by the filter. Thus, if filter 1 denies redistribution of a given route, the software does not redistribute the route, regardless of whether a filter with a higher ID permits redistribution of that route.

  NOTE: The default redistribution action is permit, even after you configure and apply a permit or deny filter. To deny redistribution of specific routes, you must configure a deny filter.

  NOTE: The option to set the metric is not applicable to static routes.

- Enable redistribution.

  NOTE: If you plan to configure redistribution filters, do not enable redistribution until you have configured the filters.

When you enable redistribution, all IP static routes are redistributed by default. If you want to deny certain routes from being redistributed into RIP, configure deny filters for those routes before you enable redistribution. You can configure up to 64 RIP redistribution filters. They are applied in ascending numerical order.

NOTE: The default redistribution action is still permit, even after you configure and apply redistribution filters to the port. If you want to tightly control redistribution, apply a filter to deny all routes as the last filter (filter ID 64), then apply filters with lower filter IDs to allow specific routes.

To configure a redistribution filter, enter a command such as the following:

```
FastIron(config-rip-router)#deny redistribute 1 static address 207.92.0.0
255.255.0.0
```

This command denies redistribution of all 207.92.x.x IP static routes.

Syntax: [no] permit | deny redistribute <filter-num> static address <ip-addr> <ip-mask> [match-metric <value> | set-metric <value>]

The <filter-num> specifies the redistribution filter ID. Specify a number from 1 – 64. The software uses the filters in ascending numerical order. Thus, if filter 1 denies a route from being redistributed, the software does not redistribute that route even if a filter with a higher ID permits redistribution of the route.

The **address** <ip-addr> <ip-mask> parameters apply redistribution to the specified network and subnet address. Use 0 to specify "any". For example, "207.92.0.0 255.255.0.0" means "any 207.92.x.x subnet". However, to specify any subnet (all subnets match the filter), enter "address 255.255.255.255 255.255.255.255".

The **match-metric** <value> parameter applies redistribution to those routes with a specific metric value; possible values are from 1 – 15.

The **set-metric** <value> parameter sets the RIP metric value that will be applied to the routes imported into RIP.

NOTE: The **set-metric** parameter does not apply to static routes.

The following command denies redistribution of a 207.92.x.x IP static route only if the route's metric is 5.

```
FastIron(config-rip-router)#deny redistribute 2 static address 207.92.0.0
255.255.0.0 match-metric 5
```

The following commands deny redistribution of all routes except routes for 10.10.10.x and 20.20.20.x:

```
FastIron(config-rip-router)#deny redistribute 64 static address 255.255.255.255
255.255.255.255
FastIron(config-rip-router)#permit redistribute 1 static address 10.10.10.0
255.255.255.0
FastIron(config-rip-router)#permit redistribute 2 static address 20.20.20.0
255.255.255.0
```

Enabling Redistribution

After you configure redistribution parameters, you need to enable redistribution.

To enable RIP redistribution, enter the following command:

```
FastIron(config-rip-router)#redistribution
```

Syntax: [no] redistribution

Enabling Learning of Default Routes

By default, the software does not learn RIP default routes.

To enable learning of default RIP routes, enter commands such as the following:

```
FastIron(config)#interface ethernet 1
FastIron(config-if-e1000-1)#ip rip learn-default
```

Syntax: interface ethernet [<slotnum>/]<portnum>

Syntax: [no] ip rip learn-default

The <slotnum>/ parameter applies to chassis devices only.

Changing the Route Loop Prevention Method

RIP can use the following methods to prevent routing loops:

- Split horizon – The Foundry device does not advertise a route on the same interface as the one on which it learned the route.
- Poison reverse – The Foundry device assigns a cost of 16 (“infinite” or “unreachable”) to a route before advertising it on the same interface as the one on which it learned the route. This is the default.

NOTE: These methods are in addition to RIP’s maximum valid route cost of 15.

To enable split horizon, enter commands such as the following:

```
FastIron(config)#interface ethernet 1
FastIron(config-if-e1000-1)#no ip rip poison-reverse
```

Syntax: [no] ip rip poison-reverse

Other Layer 3 Protocols

For information about other IP configuration commands in the Layer 2 with Base Layer 3 image that are not included in this chapter, see the chapter “Configuring IP” on page 29-1.

For information about enabling or disabling Layer 3 routing protocols, see “Enabling or Disabling Routing Protocols” on page 20-9. For complete configuration information about the routing protocols, see the other chapters in this book.

Enabling or Disabling Routing Protocols

This section describes how to enable or disable routing protocols. For complete configuration information about the routing protocols, see the other chapters in this book.

FESX and FSX devices running full Layer 3 code support the following protocols:

- BGP4
- IGMP
- IP
- IP multicast (DVMRP, PIM-SM, PIM-DM)
- OSPF
- RIPV1 and V2
- VRRP
- VRRPE
- VSRP

IP routing is enabled by default on devices running Layer 3 code. All other protocols are disabled, so you must enable them to configure and use them.

To enable a protocol on a device running full Layer 3 code, enter **router** at the global CONFIG level, followed by the protocol to be enabled. The following example shows how to enable OSPF:

```
FastIron(config)#router ospf
```

Syntax: router bgp | dvmrp | ospf | pim | rip | vrrp | vrrpe | vsrp

Enabling or Disabling Layer 2 Switching

By default, Foundry Layer 3 Switches support Layer 2 switching. These devices switch the routing protocols that are not supported on the devices. If you want to disable Layer 2 switching, you can do so globally or on individual ports, depending on the version of software your device is running.

Configuration Notes

- Make sure you really want to disable all Layer 2 switching operations before you use this option. Consult your reseller or Foundry Networks for information.
- This feature is supported in the following configurations:
  - The FESX running software release 01.1.00 or prior, supports disabling Layer 2 switching on a global basis only. Starting in release 02.1.01, the FESX supports disabling Layer 2 switching on an individual interface as well as on a global basis.
  - The FSX running software release 02.2.00 or later supports disabling Layer 2 switching on an individual interface as well as on a global basis.

Command Syntax

To globally disable Layer 2 switching on a Layer 3 Switch, enter commands such as the following:

```
FastIron(config)#route-only
FastIron(config)#exit
FastIron#write memory
FastIron#reload
```

To re-enable Layer 2 switching on a Layer 3 Switch, enter the following:

```
FastIron(config)#no route-only
FastIron(config)#exit
```

```
FastIron#write memory
FastIron#reload
```

Syntax: [no] route-only

To disable Layer 2 switching only on a specific interface, go to the Interface configuration level for that interface, then disable the feature. The following commands show how to disable Layer 2 switching on port 2:

```
FastIron(config)#interface ethernet 2
FastIron(config-if-e1000-2)#route-only
```

Syntax: [no] route-only

To re-enable Layer 2 switching, enter the command with “no”, as in the following example:

```
FastIron(config-if-e1000-2)#no route-only
```

Chapter 21
Configuring Quality of Service

Quality of Service (QoS) features are used to prioritize the use of bandwidth in a switch. When QoS features are enabled, traffic is classified as it arrives at the switch, and processed through on the basis of configured priorities. Traffic can be dropped, prioritized for guaranteed delivery, or subject to limited delivery options as configured by a number of different mechanisms.

This chapter describes how QoS is implemented and configured in the FastIron devices.

Classification

Classification is the process of selecting packets on which to perform QoS, reading the QoS information and assigning a priority to the packets. The classification process assigns a priority to packets as they enter the switch. These priorities can be determined on the basis of information contained within the packet or assigned to the packet as it arrives at the switch. Once a packet or traffic flow is classified, it is mapped to a forwarding priority queue.

Packets on Foundry's FastIron devices are classified in up to eight traffic classes with values between 0 and 7. Packets with higher priority classifications are given a precedence for forwarding.

Processing of Classified Traffic

The ***trust level*** in effect on an interface determines the type of QoS information the device uses for performing QoS. The Foundry device establishes the trust level based on the configuration of various features and if the traffic is switched or routed. The trust level can be one of the following:

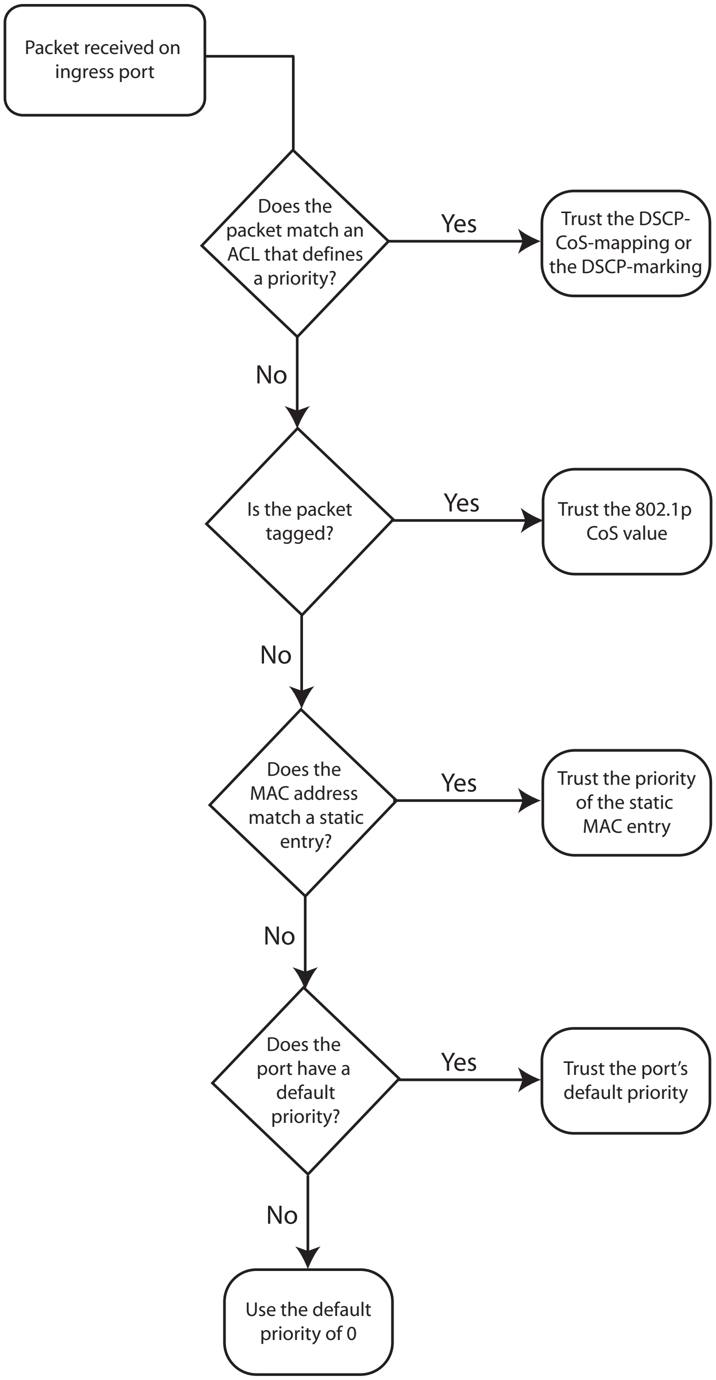
- Ingress port default priority
- Static MAC address
- Layer 2 Class of Service (CoS) value – This is the 802.1p priority value in the Ethernet frame. It can be a value from 0 – 7. The 802.1p priority is also called the Class of Service.
- Layer 3 Differentiated Service codepoint (DSCP) – This is the value in the six most significant bits of the IP packet header's 8-bit DSCP field. It can be a value from 0 – 63. These values are described in RFCs 2472 and 2475. The DSCP value is sometimes called the DiffServ value. The device automatically maps a packet's DSCP value to a hardware forwarding queue. See "Viewing QoS Settings" on page 21-13".
- ACL keyword – An ACL can also prioritize traffic and mark it before sending it along to the next hop. This is described in the ACL chapter in the section "QoS Options for IP ACLs" on page 17-27.

Given the variety of different criteria, there are multiple possibilities for traffic classification within a stream of network traffic. For this reason, the priority of packets must be resolved based on which criteria takes precedence. Precedence follows the scheme illustrated in Figure 21.1

Determining a Packet's Trust Level

Figure 21.1 illustrates how the Foundry device determines a packet's trust level.

Figure 21.1 Determining a Packet's Trust Level


Packet received on ingress port
Does the packet match an ACL that defines a priority?
Yes
Trust the DSCP-CoS-mapping or the DSCP-marking
No
Is the packet tagged?
Yes
Trust the 802.1p CoS value
No
Does the MAC address match a static entry?
Yes
Trust the priority of the static MAC entry
No
Does the port have a default priority?
Yes
Trust the port's default priority
No
Use the default priority of 0

As shown in the figure, the first criteria considered is whether the packet matches on an ACL that defines a priority. If this is not the case and the packet is tagged, the packet is classified with the 802.1p CoS value. If neither of these are true, the packet is next classified based on the static MAC address, ingress port default priority, or the default priority of zero (0).

Once a packet is classified by one of the procedures mentioned, it is mapped to an internal forwarding queue. There are eight queues designated as 0 to 7. The internal forwarding priority maps to one of these eight queues as shown in Table 21.1 through Table 21.4. The mapping between the internal priority and the forwarding queue cannot be changed.

Table 21.1 through Table 21.4 show the default QoS mappings that are used if the trust level for CoS or DSCP is enabled.

Table 21.1: Default QoS Mappings, Columns 0 to 15

DSCP value	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**12**	**14**	**15**
802.1p (COS) Value	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
DSCP value	0	1	2	3	4	5	6	7	8	9	10	11	12	12	14	15
Internal Forwarding Priority	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
Forwarding Queue	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1

Table 21.2: Default QoS Mappings, Columns 16 to 31

DSCP value	**16**	**17**	**18**	**19**	**20**	**21**	**22**	**23**	**24**	**25**	**26**	**27**	**28**	**29**	**30**	**31**
802.1p (COS) Value	2	2	2	2	2	2	2	2	3	3	3	3	3	3	3	3
DSCP value	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Internal Forwarding Priority	2	2	2	2	2	2	2	2	3	3	3	3	3	3	3	3
Forwarding Queue	2	2	2	2	2	2	2	2	3	3	3	3	3	3	3	3

Table 21.3: Default QoS Mappings, Columns 32 to 47

DSCP value	**32**	**33**	**34**	**35**	**36**	**37**	**38**	**39**	**40**	**41**	**42**	**43**	**44**	**45**	**46**	**47**
802.1p (COS) Value	4	4	4	4	4	4	4	4	5	5	5	5	5	5	5	5
DSCP value	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47
Internal Forwarding Priority	4	4	4	4	4	4	4	4	5	5	5	5	5	5	5	5
Forwarding Queue	4	4	4	4	4	4	4	4	5	5	5	5	5	5	5	5

Table 21.4: Default QoS Mappings, Columns 48 to 63

DSCP value	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63
802.1p (COS) Value	6	6	6	6	6	6	6	6	7	7	7	7	7	7	7	7
DSCP value	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63
Internal Forwarding Priority	6	6	6	6	6	6	6	6	7	7	7	7	7	7	7	7
Forwarding Queue	6	6	6	6	6	6	6	6	7	7	7	7	7	7	7	7

Mapping between DSCP value and Forwarding Queue cannot be changed. However, mapping between DSCP values and the other properties can be changed as follows:

- **DSCP to Internal Forwarding Priority Mapping** – You can change the mapping between the DSCP value and the Internal Forwarding priority value from the default values shown in Table 21.1 through Table 21.4. This mapping is used for COS marking and determining the internal priority when the trust level is DSCP. See “Changing the DSCP –> Internal Forwarding Priority Mappings” on page 21-8.
- **Internal Forwarding Priority to Forwarding Queue** – You can reassign an internal forwarding priority to a different hardware forwarding queue. See “Changing the Internal Forwarding Priority –> Hardware Forwarding Queue Mappings” on page 21-8.

QoS Queues

Foundry devices support the eight QoS queues (qosp0 – qosp7) listed in Table 21.5.

Table 21.5: QoS Queues

QoS Priority Level	QoS Queue
0	qosp0 (lowest priority queue)
1	qosp1
2	qosp2
3	qosp3
4	qosp4
5	qosp5
6	qosp6
7	qosp7 (highest priority queue)

The queue names listed above are the default names. If desired, you can rename the queues as instructed in “Renaming the Queues” on page 21-11.

Packets are classified and assigned to specific queues based on the criteria shown in Figure 21.1.

Assigning QoS Priorities to Traffic

By default, all traffic is in the best-effort queue (qosp0) and is honored on tagged ports on all FastIron family of switches. You can assign traffic to a higher queue based on the following:

- Incoming port (sometimes called the ingress port)
- Static MAC entry

The following sections describe how to change the priority for each of the items listed above.

Although it is possible for a packet to qualify for an adjusted QoS priority based on more than one of the criteria listed In the section above, the system always gives a packet the highest priority for which it qualifies. Thus, if a packet is entitled to the premium queue because of its IP source and destination addresses, but is entitled only to the high queue because of its incoming port, the system places the packet in the premium queue on the outgoing port.

When you apply a QoS priority to one of the items listed above, you specify a number from 0 – 7. The priority number specifies the IEEE 802.1 equivalent to one of the eight QoS queues on FESX, FSX, and FWSX devices. The numbers correspond to the queues as shown in Table 21.1.

Changing a Port's Priority

To change the QoS priority of port 1 to the premium queue (qosp7), enter the following commands:

```
FastIron(config)#interface ethernet 1/1
FastIron(config-if-e1000-1/1)#priority 7
```

The device will assign priority 7 to untagged switched traffic received on port 1.

Syntax: [no] priority <num>

The <num> parameter can be from 0 – 7 and specifies the IEEE 802.1 equivalent to one of eight QoS queues listed in Table 21.1.

Assigning Static MAC Entries to Priority Queues

By default, all MAC entries are in the best effort queue. When you configure a static MAC entry, you can assign the entry to a higher QoS level.

To configure a static MAC entry and assign the entry to the premium queue, enter commands such as the following:

```
FastIron(config)#vlan 9
FastIron(config-vlan-9)#static-mac-address 1145.1163.67FF ethernet 1/1 priority 7
FastIron(config-vlan-9)#write memory
```

Syntax: [no] static-mac-address <mac-addr> ethernet [<stacknum>|<slotnum>/]<portnum> [priority <num>] [host-type | router-type | fixed-host]

The <slotnum>/ parameter applies to the FSX only.

The **priority** <num> parameter can be from 0 – 7 and specifies the IEEE 802.1 equivalent to one of the eight QoS queues.

NOTE: The location of the **static-mac-address** command in the CLI depends on whether you configure port-based VLANs on the device. If the device does not have more than one port-based VLAN (VLAN 1, which is the default VLANcontaining all ports), the **static-mac-address** command is at the global CONFIG level of the CLI. If the device has more than one port-based VLAN, then the **static-mac-address** command is not available at the global CONFIG level. In this case, the command is available at the configuration level for each port-based VLAN.

Marking

Marking is the process of changing the packet's QoS information (the 802.1p and DSCP information in a packet) for the next hop. For example, for traffic coming from a device that does not support DiffServ, you can change the packet's IP Precedence value into a DSCP value before forwarding the packet.

You can mark a packet's Layer 2 CoS value, its Layer 3 DSCP value, or both values. The Layer 2 CoS or DSCP value the device marks in the packet is the same value that results from mapping the packet's QoS value into a Layer 2 CoS or DSCP value.

Marking is optional and is disabled by default. Marking is performed using ACLs. When marking is not used, the device still performs the mappings listed in “Classification” for scheduling the packet, but leaves the packet’s QoS values unchanged when the device forwards the packet.

For configuration syntax, rules, and examples of QoS marking, see “QoS Options for IP ACLs” on page 17-27.

Configuring DSCP-Based QoS

FastIron IronWare releases support basic DSCP-based QoS (also called Type of Service (ToS) based QoS) as described in this chapter. However, the FastIron family of switches do not support other advanced DSCP-based QoS features as described in the *Foundry Enterprise Configuration and Management Guide*.

Foundry FastIron IronWare releases also support marking of the DSCP value. FastIron devices can read Layer 3 Quality of Service (QoS) information in an IP packet and select a forwarding queue for the packet based on the information. The software interprets the value in the six most significant bits of the IP packet header’s 8-bit ToS field as a Diffserv Control Point (DSCP) value, and maps that value to an internal forwarding priority.

The internal forwarding priorities are mapped to one of the eight forwarding queues (qosp0 – qosp7) on the FastIron device. During a forwarding cycle, the device gives more preference to the higher numbered queues, so that more packets are forwarded from these queues. So for example, queue qosp7 receives the highest preference while queue qosp0, the best-effort queue, receives the lowest preference.

Application Notes

- DSCP-based QoS is not automatically honored for routed and switched traffic. The default is 802.1p to CoS mapping. To honor DSCP-based QoS, you must change the priority mapping to DSCP to CoS mapping. See “Using ACLs to Honor DSCP-based QoS” .
- When DSCP marking is enabled, the device changes the contents of the inbound packet’s ToS field to match the DSCP-based QoS value. This differs from the BigIron, which marks the outbound packet’s ToS field.

Using ACLs to Honor DSCP-based QoS

This section shows how to configure Foundry devices to honor DSCP-based QoS for routed and switched traffic.

FastIron GS and FastIron LS

The FastIron GS and FastIron LS support DSCP-based QoS on a per-port basis. DSCP-based QoS is not automatically honored for switched traffic. The default is 802.1p to CoS mapping. To honor DSCP-based QoS, enter the following command at the interface level of the CLI:

```
FastIron(config-if-e1000-11)trust dscp
```

When **trust dscp** is enabled, the interface honors the Layer 3 DSCP value. By default, the interface honors the Layer 2 CoS value.

FastIron X Series Devices

FastIron X Series devices require the use of an ACL to honor DSCP-based QoS for routed traffic in the Layer 3 image, or for switched traffic in the Layer 2 image. To enable DSCP-based QoS on these devices, apply an ACL entry such as the following:

```
FastIron(config)#access-list 101 permit ip any any dscp-cos-mapping
```

Configuring the QoS Mappings

You can optionally change the following QoS mappings:

- DSCP –> internal forwarding priority
- Internal forwarding priority –> hardware forwarding queue

The mappings are globally configurable and apply to all interfaces.

Default DSCP –> Internal Forwarding Priority Mappings

The DSCP values are described in RFCs 2474 and 2475. Table 21.6 list the default mappings of DSCP values to internal forwarding priority values.

Table 21.6: Default DSCP to Internal Forwarding Priority Mappings

Internal Forwarding Priority	DSCP Value
0 (lowest priority queue)	0 – 7
1	8 – 15
2	16 – 23
3	24 – 31
4	32 – 39
5	40 – 47
6	48 – 55
7 (highest priority queue)	56 – 63

Notice that DSCP values range from 0 – 63, whereas the internal forwarding priority values range from 0 – 7. Any DSCP value within a given range is mapped to the same internal forwarding priority value. For example, any DSCP value from 8 – 15 maps to priority 1.

After performing this mapping, the device maps the internal forwarding priority value to one of the hardware forwarding queues.

Table 21.7 list the default mappings of internal forwarding priority values to the hardware forwarding queues.

Table 21.7: Default Mappings of Internal Forwarding Priority Values

Internal Forwarding Priority	Forwarding Queues
0 (lowest priority queue)	qosp0
1[1]	qosp1
2	qosp2
3	qosp3
4	qosp4
5	qosp5
6	qosp6
7 (highest priority queue)	qosp7

[1] On FGS devices, you can use QoS queue 1 for priority traffic, even when sFlow is enabled on the port. This differs from the FastIron X Series. FESX, FSX, and FWSX devices with sFlow enabled support 7 priorities instead of 8 because QoS queue 1 is reserved for sFlow and not used by other packets. Any non-sFlow packets assigned to QoS queue 1 will be directed to QoS queue 0.

You can change the DSCP -> internal forwarding mappings. You also can change the internal forwarding priority -> hardware forwarding queue mappings.

Changing the DSCP –> Internal Forwarding Priority Mappings

To change the DSCP –> internal forwarding priority mappings for all the DSCP ranges, enter commands such as the following at the global CONFIG level of the CLI:

```
FastIron(config)#qos-tos map dscp-priority 0 2 3 4 to 1
FastIron(config)#qos-tos map dscp-priority 8 to 5
FastIron(config)#qos-tos map dscp-priority 16 to 4
FastIron(config)#qos-tos map dscp-priority 24 to 2
FastIron(config)#qos-tos map dscp-priority 32 to 0
FastIron(config)#qos-tos map dscp-priority 40 to 7
FastIron(config)#qos-tos map dscp-priority 48 to 3
FastIron(config)#qos-tos map dscp-priority 56 to 6
FastIron(config)#ip rebind-acl all
```

NOTE: In release 04.1.00, FastIron GS and LS devices support DCSP mapping for up to 8 values. For example: `FGS(config)#qos-tos map dscp-priority 1 2 3 4 5 6 7 8 to 6`

These commands configure the mappings displayed in the DSCP to forwarding priority portion of the QoS information display. To read this part of the display, select the first part of the DSCP value from the d1 column and select the second part of the DSCP value from the d2 row. For example, to read the DSCP to forwarding priority mapping for DSCP value 24, select 2 from the d1 column and select 4 from the d2 row. The mappings that are changed by the command above are shown below in bold type.

```
FastIron#show qos-tos
```

...portions of table omitted for simplicity...

```
DSCP-Priority map: (dscp = d1d2)

     d2|  0   1   2   3   4   5   6   7   8   9
  d1   |
  -----+----------------------------------------
   0   |  1   0   1   1   1   0   0   0   5   1
   1   |  6   1   1   1   1   1   4   2   2   2
   2   |  2   2   2   2   2   3   3   3   3   3
   3   |  3   3   0   4   4   4   4   4   4   4
   4   |  7   5   5   5   5   5   5   5   3   6
   5   |  6   6   6   6   6   6   6   7   7   7
   6   |  7   7   7   7
```

Syntax: [no] qos-tos map dscp-priority <dscp-value> [<dscp-value> ...] to <priority>

The <dscp-value> [<dscp-value> ...] parameter specifies the DSCP value ranges you are remapping. You can specify up to seven DSCP values in the same command, to map to the same forwarding priority. The first command in the example above maps priority 1 to DSCP values 0, 2, 3, and 4.

The <priority> parameter specifies the internal forwarding priority.

Changing the Internal Forwarding Priority –> Hardware Forwarding Queue Mappings

To reassign an internal forwarding priority to a different hardware forwarding queue, enter commands such as the following at the global CONFIG level of the CLI:

```
FastIron(config)#qos tagged-priority 2 qosp0
```

Syntax: [no] qos tagged-priority <num> <queue>

The <num> parameter can be from 0 – 7 and specifies the internal forwarding priority.

The <queue> parameter specifies the hardware forwarding queue to which you are reassigning the priority. The default queue names are as follows:

- qosp7
- qosp6
- qosp5
- qosp4
- qosp3
- qosp2
- qosp1
- qosp0

Scheduling

Scheduling is the process of mapping a packet to an internal forwarding queue based on its QoS information, and servicing the queues according to a mechanism.

This section describes the scheduling methods used on FESX, FSX, FWSX, FGS and FLS devices.

QoS Queuing Methods

The following QoS queuing methods are supported in all IronWare releases for the FastIron FESX, FSX, FWSX, FGS and FLS devices.

- ***Weighted Round Robin (WRR)*** – WRR ensures that all queues are serviced during each cycle. A weighted fair queuing algorithm is used to rotate service among the eight queues on FESX, FSX, FWSX, FGS and FLS devices. The rotation is based on the weights you assign to each queue. This method rotates service among the queues, forwarding a specific number of packets in one queue before moving on to the next one.

  WRR is the default queuing method and uses a default set of queue weights.

  The number of packets serviced during each visit to a queue depends on the percentages you configure for the queues. The software automatically converts the percentages you specify into weights for the queues.

  NOTE: Queue cycles on the FESX, FSX, FWSX, FGS and FLS devices are based on bytes. These devices service a given number of bytes (based on weight) in each queue cycle. FES and BI/FI queue cycles are based on packets. The bytes-based scheme is more accurate than a packets-based scheme if packets vary greatly in size.

- ***Strict Priority(SP)*** – SP ensures service for high priority traffic. The software assigns the maximum weights to each queue, to cause the queuing mechanism to serve as many packets in one queue as possible before moving to a lower queue. This method biases the queuing mechanism to favor the higher queues over the lower queues.

  For example, strict queuing processes as many packets as possible in qosp3 before processing any packets in qosp2, then processes as many packets as possible in qosp2 before processing any packets in qosp1, and so on.

- ***Hybrid WRR and SP*** – Starting with software release 02.2.00, an additional configurable queueing mechanism combines both the strict priority and weighted round robin mechanisms. The combined method enables the Foundry device to give strict priority to delay-sensitive traffic such as VOIP traffic, and weighted round robin priority to other traffic types.

By default, when you select the combined SP and WRR queueing method, the Foundry device assigns strict priority to traffic in qosp7 and qosp6, and weighted round robin priority to traffic in qosp0 through qosp5. Thus, the Foundry device schedules traffic in queue 7 and queue 6 first, based on the strict priority queueing method. When there is no traffic in queue 7 and queue 6, the device schedules the other queues in round-robin fashion from the highest priority queue to the lowest priority queue.

By default, when you specify the combined SP and WRR queuing method, the system balances the traffic among the queues as shown in Table 21.8. If desired, you can change the default bandwidth values as instructed in the section “Changing the Bandwidth Allocations of the Hybrid WRR and SP Queues” on page 21-12.

Table 21.8: Default Bandwidth for Combined SP and WRR Queueing Methods

Queue	Default Bandwidth
qosp7	Strict priority (highest priority)
qosp6	Strict priority
qosp5	25%
qosp4	15%
qosp3	15%
qosp2	15%
qosp1	15%
qosp0	15% (lowest priority)

Selecting the QoS Queuing Method

By default, FastIron devices use the weighted fair queuing method of packet prioritization. To change the method to strict priority or back to weighted fair queuing, enter the following command at the Global CONFIG level of the CLI:

```
FastIron(config)#qos mechanism strict
```

To change the method back to weighted round robin, enter the following command:

```
FastIron(config)#qos mechanism weighted
```

Syntax: [no] qos mechanism strict | weighted

NOTE: The following combined method is supported in releases 02.2.00 and later.

To change the queuing mechanism to the combined SP and WRR method, enter the following command at the Global CONFIG level of the CLI:

```
FastIron(config)#qos mechanism mixed-sp-wrr
```

Syntax: mechanism mixed-sp-wrr

Configuring the QoS Queues

Each of the queues has the following configurable parameters:

- The queue name
- The minimum percentage of a port’s outbound bandwidth guaranteed to the queue

Renaming the Queues

The default queue names on FESX, FSX, FWSX, FGS and FLS devices are qosp7, qosp6, qosp5, qosp4, qosp3, qosp2, qosp1, and qosp0. You can change one or more of the names if desired.

To rename queue qosp3 to "92-octane", enter the following commands:

```
FastIron(config)#qos name qosp3 92-octane
```

Syntax: qos name <old-name> <new-name>

The <old-name> parameter specifies the name of the queue before the change.

The <new-name> parameter specifies the new name of the queue. You can specify an alphanumeric string up to 32 characters long.

Changing the Minimum Bandwidth Percentages of the WRR Queues

If you are using the weighted round robin mechanism instead of the strict mechanism, you can change the weights for each queue by changing the minimum percentage of bandwidth you want each queue to guarantee for its traffic.

By default, the eight QoS queues on FESX, FSX, FWSX, FGS and FLS devices receive the following minimum guaranteed percentages of a port's total bandwidth. Note that the defaults differ when jumbo frames are enabled.

Table 21.9: Default Minimum Bandwidth Percentages on FESX, FSX, FWSX, FGS and FLS devices

Queue	Default Minimum Percentage of Bandwidth	
	Without Jumbo Frames	**With Jumbo Frames**
qosp7	75%	44%
qosp6	7%	8%
qosp5	3%	8%
qosp4	3%	8%
qosp3	3%	8%
qosp2	3%	8%
qosp1	3%	8%
qosp0	3%	8%

When the queuing method is weighted round robin, the software internally translates the percentages into weights. The weight associated with each queue controls how many packets are processed for the queue at a given stage of a cycle through the weighted round robin algorithm.

NOTE: Queue cycles on the FESX, FSX, FWSX, FGS and FLS devices are based on bytes. These devices service a given number of bytes (based on the weight) in each queue cycle. FES and BI/FI queue cycles are based on packets. The bytes-based scheme is more accurate than a packets-based scheme if packets vary greatly in size.

The bandwidth allocated to each queue is based on the relative weights of the queues. You can change the bandwidth percentages allocated to the queues by changing the queue weights.

There is no minimum bandwidth requirement for a given queue. For example, queue qosp3 is not required to have at least 50% of the bandwidth.

Command Syntax

To change the bandwidth percentages for the queues, enter commands such as the following. Note that this example uses the default queue names.

```
FastIron(config)#qos profile qosp7 25 qosp6 15 qosp5 12 qosp4 12 qosp3 10  qosp2 10
qosp1 10 qosp0 6
Profile qosp7     : Priority7   bandwidth requested  25% calculated  25%
Profile qosp6     : Priority6   bandwidth requested  15% calculated  15%
Profile qosp5     : Priority5   bandwidth requested  12% calculated  12%
Profile qosp4     : Priority4   bandwidth requested  12% calculated  12%
Profile qosp3     : Priority3   bandwidth requested  10% calculated  10%
Profile qosp2     : Priority2   bandwidth requested  10% calculated  10%
Profile qosp1     : Priority1   bandwidth requested  10% calculated  10%
Profile qosp0     : Priority0   bandwidth requested   6% calculated   6%
```

Syntax: [no] qos profile <queue> <percentage> <queue> <percentage> <queue> <percentage> <queue> <percentage> <queue> <percentage> <queue> <percentage> <queue> <percentage> <queue> <percentage>

Each <queue> parameter specifies the name of a queue. You can specify the queues in any order on the command line, but you must specify each queue.

The <percentage> parameter specifies a number for the percentage of the device’s outbound bandwidth that is allocated to the queue. FESX, FSX, FWSX, FGS and FLS QoS queues require a minimum bandwidth percentage of 3% for each priority. When jumbo frames are enabled, the minimum bandwidth requirement is 8%. If these minimum values are not met, QoS may not be accurate.

Configuration Notes

- The total of the percentages you enter must equal 100.
- FESX, FSX, FWSX, FGS and FLS devices do not adjust the bandwidth percentages you enter. BigIron QoS does adjust the bandwidth percentages to ensure that each queue has at least its required minimum bandwidth percentage.

On the FastIron GS, you can use QoS queue 1 for priority traffic, even when sFlow is enabled on the port. This differs from the FastIron X Series. When sFlow is enabled on the FESX, FSX, and FWSX, these devices support seven priorities instead of eight because QoS queue 1 is reserved for sFlow and is not used by other packets. Any non-sFlow packets assigned to QoS queue 1 will be directed to QoS queue 0.

Changing the Bandwidth Allocations of the Hybrid WRR and SP Queues

Platform Support:

- FESX/FSX/FWSX devices running software release 02.2.00 and later

To change the default bandwidth percentages for the queues when the device is configured to use the combined SP and WRR queuing mechanism, enter commands such as the following. Note that this example uses the default queue names.

```
FastIron(config)#qos profile qosp7 sp qosp6 sp qosp5 20 qosp4 16 qosp3 16 qosp2 16
qosp1 16 qosp0 16
```

Syntax: [no] qos profile <queue 7> sp <queue 6> sp | <percentage> <queue 5> <percentage> <queue 4> <percentage> <queue 3> <percentage> <queue 2> <percentage> <queue 1> <percentage> <queue 0> <percentage>]

Each **<queue *x*>** parameter specifies the name of a queue. You can specify the queues in any order on the command line, but you must specify each queue. Note that queue 7 supports strict priority only, queue 6 supports both strict priority and WRR queuing mechanisms, and queues 0 – 5 support the WRR queuing mechanism only.

The **sp** parameter configures strict priority as the queuing mechanism. Note that only queue 7 and queue 6 support this method.

The **<percentage>** parameter configures WRR as the queuing mechanism and specifies the percentage of the device's outbound bandwidth allocated to the queue. The queues require a minimum bandwidth percentage of 3% for each priority. When jumbo frames are enabled, the minimum bandwidth requirement is 8%. If these minimum values are not met, QoS may not be accurate.

NOTE: The percentages must add up to 100. The Foundry device does not adjust the bandwidth percentages you enter. In contrast, the BigIron QoS does adjust the bandwidth percentages to ensure that each queue has at least its required minimum bandwidth percentage.

Viewing QoS Settings

To display the QoS settings for all the queues, enter the **show qos-profiles** command, as shown in the following examples.

The following shows an example display output on a FESX.

```
FastIron#show qos-profiles all
bandwidth scheduling mechanism: weighted priority
Profile qosp7     : Priority7   bandwidth requested  25% calculated  25%
Profile qosp6     : Priority6   bandwidth requested  15% calculated  15%
Profile qosp5     : Priority5   bandwidth requested  12% calculated  12%
Profile qosp4     : Priority4   bandwidth requested  12% calculated  12%
Profile qosp3     : Priority3   bandwidth requested  10% calculated  10%
Profile qosp2     : Priority2   bandwidth requested  10% calculated  10%
Profile qosp1     : Priority1   bandwidth requested  10% calculated  10%
Profile qosp0     : Priority0   bandwidth requested   6% calculated   6%
```

Syntax: show qos-profiles all | <name>

The **all** parameter displays the settings for all eight queues.

The <name> parameter displays the settings for the specified queue.

Viewing DSCP-based QoS Settings

To display configuration information for DSCP-based QoS, enter the following command at any level of the CLI:

```
FastIron#show qos-tos
DSCP-->Traffic-Class map: (DSCP = d1d2: 00, 01...63)
      d2|  0   1   2   3   4   5   6   7   8   9
  d1   |
  -----+----------------------------------------
   0   |  0   0   0   0   0   0   0   0   1   1
   1   |  1   1   1   1   1   1   2   2   2   2
   2   |  2   2   2   2   3   3   3   3   3   3
   3   |  3   3   4   4   4   4   4   4   4   4
   4   |  5   5   5   5   5   5   5   5   6   6
   5   |  6   6   6   6   6   6   7   7   7   7
   6   |  7   7   7   7

Traffic-Class-->802.1p-Priority map (use to derive DSCP--802.1p-Priority):
Traffic | 802.1p
Class   | Priority
--------+---------
   0    |    0
   1    |    1
   2    |    2
   3    |    3
   4    |    4
   5    |    5
   6    |    6
   7    |    7
--------+---------
```

Syntax: show qos-tos

This command shows the following information.

Table 21.10: DSCP-based QoS Configuration Information

This Field...	Displays...
DSCP-Priority map	
d1 and d2	The DSCP to forwarding priority mappings that are currently in effect. **Note**: The example above shows the default mappings. If you change the mappings, the command displays the changed mappings
Traffic Class -> 802.1 Priority map	
Traffic Class and 802.1p Priority	The traffic class to 802.1p Priority mappings that are currently in effect. **Note**: The example above shows the default mappings. If you change the mappings, the command displays the changed mappings.

Chapter 22

Configuring Rate Limiting and Rate Shaping on FastIron X Series Switches

This chapter describes how to configure rate limiting and rate shaping on Foundry's FESX, FSX, and FWSX devices.

Rate limiting applies to inbound ports and rate shaping applies to outbound ports.

NOTE: This chapter applies to the FastIron X Series switches. To configure rate limiting on the FastIron GS and FastIron LS, see the chapter "Configuring Rate Limiting on the FastIron GS and FastIron LS" on page 23-1".

Overview

Foundry's FastIron X Series devices support port-based fixed rate limiting on inbound ports. Fixed Rate Limiting allows you to specify the maximum number of bytes a given port can receive. The port drops bytes that exceed the limit you specify. You can configure a Fixed Rate Limiting policy on a port's inbound direction only. Fixed rate limiting applies to all traffic on the rate limited port.

Fixed rate limiting on FastIron X-Series devices is at line rate and occurs in hardware. See "Rate Limiting in Hardware" on page 15-2.

When you specify the maximum number of bytes, you specify it in bits per second (bps). The Fixed Rate Limiting policy applies to one-second intervals and allows the port to receive the number of bytes you specify in the policy, but drops additional bytes. Unused bandwidth is not carried over from one interval to the next.

NOTE: Foundry recommends that you do not use Fixed Rate Limiting on ports that receive route control traffic or Spanning Tree Protocol (STP) control traffic. If the port drops control packets due to the Fixed Rate Limiting policy, routing or STP can be disrupted.

Rate Limiting in Hardware

Each FastIron X Series device supports line-rate rate limiting in hardware. The device creates entries in Content Addressable Memory (CAM) for the rate limiting policies. The CAM entries enable the device to perform the rate limiting in hardware instead of sending the traffic to the CPU. The device sends the first packet in a given traffic flow to the CPU, which creates a CAM entry for the traffic flow. A CAM entry consists of the source and destination addresses of the traffic. The device uses the CAM entry for rate limiting all the traffic within the same flow. A rate limiting CAM entry remains in the CAM for two minutes before aging out.

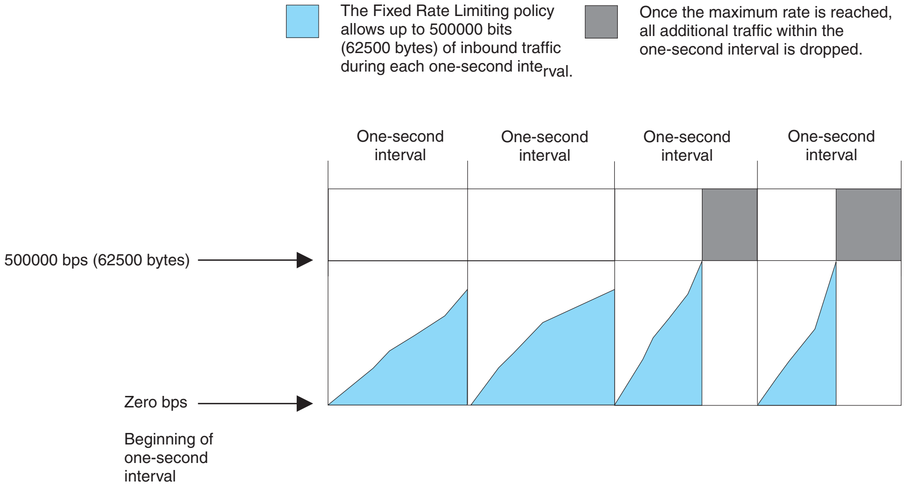
How Fixed Rate Limiting Works

Fixed Rate Limiting counts the number of bytes that a port receives, in one second intervals. If the number of bytes exceeds the maximum number you specify when you configure the rate, the port drops all further inbound packets for the duration of the one-second interval.

Once the one-second interval is complete, the port clears the counter and re-enables traffic.

Figure 15.1 shows an example of how Fixed Rate Limiting works. In this example, a Fixed Rate Limiting policy is applied to a port to limit the inbound traffic to 500000 bits (62500 bytes) a second. During the first two one-second intervals, the port receives less than 500000 bits in each interval. However, the port receives more than 500000 bits during the third and fourth one-second intervals, and consequently drops the excess traffic.

Figure 22.1 Fixed Rate Limiting


The Fixed Rate Limiting policy allows up to 500000 bits (62500 bytes) of inbound traffic during each one-second interval.
Once the maximum rate is reached, all additional traffic within the one-second interval is dropped.
One-second interval
One-second interval
One-second interval
One-second interval
500000 bps (62500 bytes)
Zero bps
Beginning of one-second interval

NOTE: The software counts the bytes by polling statistics counters for the port every 100 milliseconds, which provides 10 readings each second. Due to the polling interval, the Fixed Rate Limiting policy has an accuracy of within 10% of the port's line rate. It is therefore possible for the policy to sometimes allow more traffic than the limit you specify, but the extra traffic is never more than 10% of the port's line rate.

Configuration Notes

- Rate limiting is available only on inbound ports on FastIron X Series devices.
- Fixed rate limiting is not supported on 10-Gigabit Ethernet ports.
- Fixed rate limiting is not supported on tagged ports in the full Layer 3 router image.

Configuring a Port-Based Rate Limiting Policy

To configure rate limiting on a port, enter commands such as the following:

```
FastIron(config)#interface ethernet 24
FastIron(config-if-e1000-24)#rate input fixed 500000
```

These commands configure a fixed rate limiting policy that allows port 24 to receive a maximum of 500000 bits per second (62500 bytes per second). If the port receives additional bytes during a given one-second interval, the port drops all inbound packets on the port until the next one-second interval starts.

Syntax: [no] rate-limit input fixed <average-rate>

The <average-rate> parameter specifies the maximum number of bits per second (bps) the port can receive. The minimum rate that can be configured on FESX, FSX, and FWSX devices is 64,000 bps.

Configuring an ACL-Based Rate Limiting Policy

Platform Support:

- FESX/FSX/FWSX devices running software release 02.3.03 and later

IP ACL-based rate limiting of inbound traffic provides the facility to limit the rate for IP traffic that matches the permit conditions in extended IP ACLs. This feature is available in the Layer 2 and Layer 3 code.

To configure ACL-based rate limiting on an X Series device, you create individual ***traffic policies***, then reference the traffic policies in one or more ACL entries (also called clauses or statements). The traffic policies become effective on ports to which the ACLs are bound.

For configuration procedures for ACL-based rate limiting see the chapter "Configuring Traffic Policies" on page 24-1.

Optimizing Rate Limiting

By default, rate limiting is optimized for packets that are 256 bytes in size. This packet size includes 14 bytes of Layer 2 header (Ethernet II untagged) and 4 bytes of Layer 2 CRC.

To optimize rate limiting for all packet sizes, use the **payload-only** parameter. When this parameter is specified, the system excludes Layer 2 header and Layer 2 checksum (CRC) from the calculations, and the rate is accurate for all packet sizes and Layer 2 overhead (Layer 2 header + CRC). Layer 2 overhead for different encapsulations is as follows:

- Untagged Ethernet-II – 18 bytes
- Tagged Ethernet-II – 22 bytes
- LLC over Untagged Ethernet-II – 21 bytes
- LLC over Tagged Ethernet-II – 25 bytes
- LLC/SNAP over Untagged Ethernet-II – 26 bytes
- LLC/SNAP over Tagged Ethernet-II – 30 bytes

To optimize rate limiting, enter commands such as the following:

```
FastIron(config)#interface ethernet 24
FastIron(config-if-e1000-24)#rate input fixed 500000 payload-only
```

These commands configure a fixed rate limiting policy that allows port 24 to receive a maximum of 500000 bits per second. The payload-only parameter causes the device to exclude the Layer 2 header and Layer 2 checksum from the calculations.

NOTE: When you enable the **payload-only** parameter on the FESX, FSX, and FWSX devices, the configuration applies to all the other ports in the same port region. For example, if you enable the **payload-only** option on port 12 on a FESX424, the configuration applies to ports 1 through 12 since these ports are in the same port region.

Syntax: [no] rate-limit input fixed <average-rate> [payload-only]

The <average-rate> parameter specifies the maximum number of bits per second (bps) the port can receive. The minimum rate that can be configured on FESX, FSX, and FWSX devices is 64,000 bps. By default, rate limiting is optimized for packets that are 256 bytes in size.

Displaying the Fixed Rate Limiting Configuration

To display the fixed rate limiting configuration on the device, enter the following command:

```
FastIron#show rate-limit fixed
Total rate-limited interface count: 11.
 Port    Configured Input Rate       Actual Input Rate            Mode
    1                  1000000                 1000000    Payload-Only
    3                 10000000                10005000         Default
    7                 10000000                10000000    Payload-Only
    9                  7500000                 7502000    Payload-Only
   11                  8000000                 7999000         Default
   12                  8000000                 7999000         Default
   13                  8000000                 7999000         Default
   14                  8000000                 7999000         Default
   15                  8000000                 7999000         Default
   21                  8000000                 8000000    Payload-Only
   25                  7500000                 7502000         Default
```

Syntax: show rate-limit fixed

The command lists the ports on which fixed rate limiting is configured, and provides the information listed in Table 15.1 for each of the ports.

Table 22.1: CLI Display of Fixed Rate Limiting Information

This Field...	Displays...
Total rate-limited interface count	The total number of ports that are configured for Fixed Rate Limiting.
Port	The port number.
Configured Input Rate	The maximum rate requested for inbound traffic. The rate is measured in bits per second (bps).
Actual Input Rate	The actual maximum rate provided by the hardware. The rate is measured in bps.

Rate Shaping

Platform Support:

- FESX/FSX/FWSX devices running software release 03.0.00 and later

Outbound Rate Shaping is a port level feature that is used to shape the rate and to control the bandwidth of outbound traffic on a port. This feature smooths out excess and bursty traffic to the configured maximum limit before it is sent out on a port. Packets are stored in available buffers and then forwarded at a rate no greater than the configured limit. This process provides for better control over the inbound traffic of neighboring devices.

The device has one global rate shaper for a port and one rate shaper for each port priority queue. Rate shaping is done on a single-token basis, where each token is defined to be 1 byte.

The following rules apply when configuring outbound rate shapers:

- Outbound rate shapers can be configured *only* on physical ports, not on virtual or loopback ports.
- For trunk ports, the rate shaper must be configured on individual ports of a trunk using the **config-trunk-ind** command (trunk configuration level); you cannot configure a rate shaper for a trunk.
- You can configure a rate shaper for a port and for the individual priority queues of that port. However, if a port

rate shaper is configured, that value overrides the rate shaper value of a priority queue if the priority queue's rate shaper is greater than the rate shaper for the port.

- Configured rate shaper values are rounded up to the nearest multiple of 651 Kbps. The maximum configurable limit is 2665845 Kbps.

Configuring Outbound Rate Shaping for a Port

To configure the maximum rate at which outbound traffic is sent out on a port, enter the following:

```
FastIron(config)#interface e 1/2
FastIron(config-if-e1000-2)#rate-limit output shaping 1300
```

The configured 1300 Kbps outbound rate shaping on Port 2 is rounded up to the nearest multiple of 651 Kbps, which is 1302 Kbps. This value is the actual limit on the port for outbound traffic.

Syntax: [no] rate-limit output shaping <value>

You can configure up to 2665845 Kbps for <value>.

Configuring Outbound Rate Shaping for a Specific Priority

To configure the maximum rate at which outbound traffic is sent out on a port's priority queue, enter the following:

```
FastIron(config)#interface e 1/2
FastIron(config-if-e1000-1)#rate-limit output shaping 500 priority 7
```

The configured 500 Kbps limit for outbound traffic on Priority queue 7 on Port 2 is rounded up to the nearest multiple of 651 Kbps, which is 651 Kbps.

Syntax: [no] rate-limit output shaping <value> priority <priority-queue>

You can configure up to 2665845 Kbps for <value>.

Specify 0-7 for <priority-queue>

Configuring Outbound Rate Shaping for a Trunk Port

To configure the maximum rate at which outbound traffic is sent out on a trunk port, enter the following on each trunk port where outbound traffic will be shaped.

```
FastIron(config)#trunk e 1/13 to 1/16
FastIron(config-trunk-13-16)#config-trunk-ind
FastIron(config-trunk-13-16)#rate-limit output shaping ethe 1/15 651
FastIron(config-trunk-13-16)#rate-limit output shaping ethe 1/14 1300
```

An outbound rate shaper is configured on Port 1/14 and Port 1/15. The configured outbound rate shaper (651 Kbps) on Port 1/15 is the maximum rate of outbound traffic that is sent out on that port, since 651 Kbps is a multiple of 651 Kbps.

The configured 1300 Kbps limit on Port 14 is rounded up to 1302 Kbps.

Syntax: [no] rate-limit output shaping ethernet [<slotnum>/]<portnum> <value>

The <slotnum> parameter is required on chassis devices.

You can configure up to 2665845 Kbps for <value>.

Displaying Rate Shaping Configurations

To display the configured outbound rate shaper on a device, enter the following command:

```
FastIron#show rate-limit output-shaping
Outbound Rate Shaping Limits in Kbps:
 Port   PortMax   Prio0   Prio1   Prio2   Prio3   Prio4   Prio5   Prio6   Prio7
    1         -       -       -       -       -       -       -       -     651
    2      1302       -       -       -       -       -       -       -       -
   15       651       -       -       -       -       -       -       -       -
```

```
show rate-limit output-shaping
```

The display lists the ports on a device, the configured outbound rate shaper on a port and for a priority for a port.

Chapter 23

Configuring Rate Limiting on the FastIron GS and FastIron LS

This chapter describes how to configure fixed rate limiting on inbound and outbound ports on Foundry's FastIron GS and LS models using the CLI.

NOTE: This chapter applies to the FastIron GS and FastIron LS models. To configure rate limiting on a FastIron X Series device, see "Configuring Rate Limiting and Rate Shaping on FastIron X Series Switches" on page 22-1.

Overview

Foundry's FastIron GS and FastIron LS devices support port-based fixed rate limiting on inbound ports and outbound ports. Fixed Rate Limiting allows you to specify the maximum number of bytes a given port can send or receive. The port drops bytes that exceed the limit you specify. You can configure a Fixed Rate Limiting policy on a port's inbound or outbound direction. Fixed rate limiting applies to all traffic on the rate limited port.

When you specify the maximum number of bytes, you specify it in kilobits per second (Kbps). The Fixed Rate Limiting policy applies to one-second intervals and allows the port to send or receive the number of bytes you specify in the policy, but drops additional bytes. Unused bandwidth is not carried over from one interval to the next.

NOTE: Foundry recommends that you do not use Fixed Rate Limiting on ports that send or receive route control traffic or Spanning Tree Protocol (STP) control traffic. If the port drops control packets due to the Fixed Rate Limiting policy, routing or STP can be disrupted.

Rate Limiting in Hardware

Each FastIron GS and FastIron LS device supports line-rate rate limiting in hardware. The device creates entries in Content Addressable Memory (CAM) for the rate limiting policies. The CAM entries enable the device to perform the rate limiting in hardware instead of sending the traffic to the CPU. The device sends the first packet in a given traffic flow to the CPU, which creates a CAM entry for the traffic flow. A CAM entry consists of the source and destination addresses of the traffic. The device uses the CAM entry for rate limiting all the traffic within the same flow. A rate limiting CAM entry remains in the CAM for two minutes before aging out.

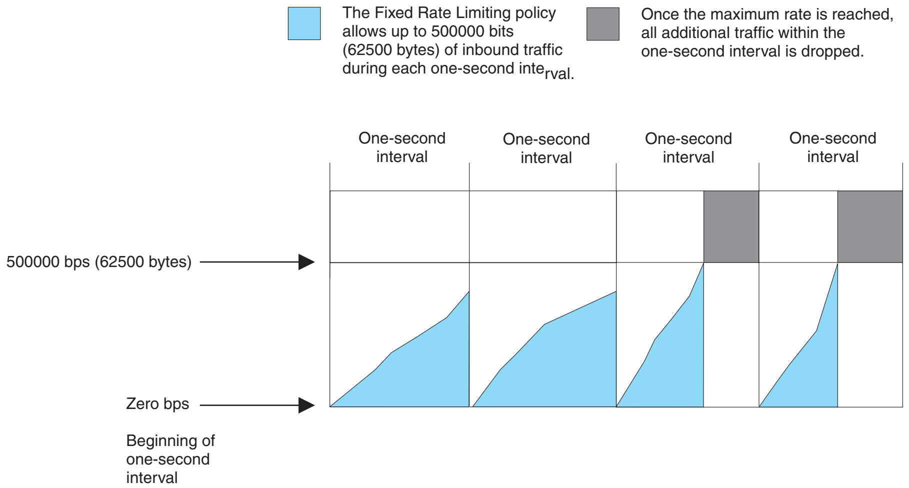
How Fixed Rate Limiting Works

Fixed Rate Limiting counts the number of bytes that a port either sends or receives, in one second intervals. The direction that the software monitors depends on the direction you specify when you configure the rate limit on the port. If the number of bytes exceeds the maximum number you specify when you configure the rate, the port drops all further packets for the rate-limited direction, for the duration of the one-second interval.

Once the one-second interval is complete, the port clears the counter and re-enables traffic.

Figure 23.1 shows an example of how Fixed Rate Limiting works. In this example, a Fixed Rate Limiting policy is applied to a port to limit the inbound traffic to 500000 bits (62500 bytes) a second. During the first two one-second intervals, the port receives less than 500000 bits in each interval. However, the port receives more than 500000 bits during the third and fourth one-second intervals, and consequently drops the excess traffic.

Figure 23.1 Fixed Rate Limiting


The Fixed Rate Limiting policy allows up to 500000 bits (62500 bytes) of inbound traffic during each one-second interval.
Once the maximum rate is reached, all additional traffic within the one-second interval is dropped.
One-second interval
One-second interval
One-second interval
One-second interval
500000 bps (62500 bytes)
Zero bps
Beginning of one-second interval

NOTE: The software counts the bytes by polling statistics counters for the port every 100 milliseconds, which provides 10 readings each second. Due to the polling interval, the Fixed Rate Limiting policy has an accuracy of within 10% of the port's line rate. It is therefore possible for the policy to sometimes allow more traffic than the limit you specify, but the extra traffic is never more than 10% of the port's line rate.

Configuring Fixed Rate Limiting on Inbound Ports

Inbound rate limiting allows you to specify the maximum number of Kbps a given port can receive.

Minimum and Maximum Rates

Table 23.1 lists the minimum and maximum inbound rate limits on GbE and 10-GbE ports

.

Table 23.1: Rates for Inbound Rate Limiting

Port Type	Minimum Rate	Maximum Rate
GbE	65 Kbps	1000000 Kbps
10-GbE	65 Kbps	10000000 Kbps

Configuration Notes

- Inbound rate limiting is supported on:
    - GbE ports
    - 10-GbE ports
    - Trunk ports

- Inbound rate limiting is not supported on:
    - Ports on which LACP is enabled
    - Virtual interfaces
    - Loopback interfaces

Configuration Syntax

To configure inbound rate limiting on a port, enter commands such as the following:

```
FastIron(config)#interface ethernet 0/2/1
FastIron(config-if-e10000-0/2/1)#rate-limit input fixed 1000000
Rate Limiting on Port 0/2/1 - Config: 1000000 Kbps, Actual: 1000000 Kbps
```

The above commands configure a fixed rate limiting policy that allows port 0/2/1, a 10-GbE port, to receive a maximum of 1000000 kilobits per second. If the port receives additional bits during a given one-second interval, the port drops all inbound packets on the port until the next one-second interval starts.

```
FastIron(config)#interface ethernet 0/1/10
FastIron(config-if-e1000-0/1/10)#rate-limit input fixed 1000
Rate Limiting on Port 0/1/10 - Config: 1000 Kbps, Actual: 1000 Kbps
```

The above commands configure a fixed rate limiting policy that allows port 0/1/10, a GbE port, to receive a maximum of 1000 kilobits per second. If the port receives additional bits during a given one-second interval, the port drops all inbound packets on the port until the next one-second interval starts.

Syntax: [no] rate-limit input fixed <average-rate>

The <average-rate> parameter specifies the maximum number of kilobits per second (Kbps) the port can receive. Table 23.1 lists the minimum and maximum rates.

Configuring Fixed Rate Limiting on Outbound Ports

Outbound rate limiting allows you to specify the maximum number of kilobits a given port can transmit.

The FastIron GS and FastIron LS support the following types of outbound fixed rate limiting on Gigabit and 10-Gigabit Ethernet ports.

- **Port-based** – Limits the rate of outbound traffic on an individual physical port or trunk port, to a specified rate. Traffic that exceeds the maximum rate is dropped. Only one port-based outbound rate limiting policy can be applied to a port.
- **Port- and priority-based** – Limits the rate on an individual 802.1p priority queue on an individual physical port or trunk port. Traffic that exceeds the rate is dropped. Only one priority-based rate limiting policy can be specified per priority queue for a port. This means that a maximum of eight port- and priority-based policies can be configured on a port.

Minimum and Maximum Rates

Table 23.2 lists the minimum and maximum outbound rate limits on GbE and 10-GbE ports.

Table 23.2: Rates for Outbound Rate Limiting

Port Type	Minimum Rate	Maximum Rate	Granularity
GbE	65 Kbps	1000000 Kbps	65 Kbps
10-GbE	2500 Kbps	10000000 Kbps	2500 Kbps

Configuration Notes

- Outbound rate limiting is supported on:
    - GbE ports
    - 10-GbE ports
    - Trunk ports
- Outbound rate limiting is not supported on:
    - Ports on which LACP is enabled
    - Virtual interfaces
    - Loopback interfaces
- Because of the hardware architecture of the FastIron GS and FastIron LS, the effect of outbound rate limiting differs on GbE ports compared to 10-GbE ports. For example, applying the same rate limiting value on GbE and 10-GbE ports will produce different results.
- You can configure both outbound port-base rate limiting and outbound port- and priority-based rate limiting on a single physical port or trunk port. However, if a priority-based limit for a given port is greater than the port-based rate limit, then the port-based rate limit will override the priority-based rate limit. Similarly, if the port-based rate limit is greater than the priority-based limit, then the priority-based rate limit will override the port-based rate limit.

Port-Based Rate Limiting

To configure port-based fixed rate limiting on an outbound port, enter commands such as the following:

```
FastIron(config)#interface ethernet 0/1/34
FastIron(config-if-e1000-0/1/34)#rate-limit output fixed 32
Outbound Rate Limiting on Port 0/1/34 Config: 32 Kbps, Actual: 65 Kbps
```

The above commands configure a fixed rate limiting policy that allows port 0/1/34 to transmit 32 Kbps. Since port 0/1/34 is a GbE port and the minimum rate is 65 Kbps (see Table 23.2), the system will adjust the configured rate of 32 Kbps to an actual rate to 65 Kbps. If the port transmits additional bits during a given one-second interval, the port will drop all outbound packets on the port until the next one-second interval starts.

```
FastIron(config)#interface ethernet 0/2/1
FastIron(config-if-e1000-0/2/1)#rate-limit output fixed 32
Outbound Rate Limiting on Port 0/2/1 Config: 32 Kbps, Actual: 2500 Kbps
```

The above commands configure a fixed rate limiting policy that allows port `0/2/1` to transmit 32 Kbps per second. Since port `0/2/1` is a 10-GbE port and the minimum rate is 2500 Kbps (see Table 23.2), the system will adjust the configured rate of 32 Kbps to an actual rate of 2500 Kbps. If the port transmits additional bits during a given one-second interval, the port will drop all outbound packets on the port until the next one-second interval starts.

Syntax: [no] rate-limit output fixed <average-rate>

The <average-rate> parameter specifies the average number of kilobits per second (Kbps) the port can send. Table 23.2 lists the minimum and maximum rates for GbE and 10-GbE ports.

Port- and Priority-Based Rate Limiting

Port- and priority-based rate limiting limits the rate on an individual 802.1p priority queue on an individual physical port or trunk port.

To configure port- and priority-based fixed rate limiting on an outbound port, enter commands such as the following:

```
FastIron(config)#interface ethernet 0/1/35
FastIron(config-if-e1000-0/1/35)#rate-limit output fixed 1000 priority 7
Outbound Rate Limiting on Port 0/1/35 for Priority 7
  Config: 1000 Kbps, Actual: 975 Kbps
```

The above commands configure a fixed rate limiting policy that allows traffic with a priority of 7 on port 0/1/35 to transmit 1000 Kbps per second. The system rounds the configured rate to 975 Kbps. If the port transmits additional bits during a given one-second interval, the port will drop all outbound packets on the port until the next one-second interval starts.

Configuring an ACL-Based Rate Limiting Policy

The FastIron GS and FastIron LS devices support IP ACL-based rate limiting of inbound traffic. ACL-based rate limiting provides the facility to limit the rate for IP traffic that matches the permit conditions in extended IP ACLs.

To configure ACL-based rate limiting, you create individual ***traffic policies***, then reference the traffic policies in one or more ACL entries (also called clauses or statements). The traffic policies become effective on ports to which the ACLs are bound.

For configuration procedures for ACL-based rate limiting, see “Configuring Traffic Policies” on page 24-1.

Displaying the Fixed Rate Limiting Configuration

You can display the fixed rate limiting configuration for inbound and outbound traffic.

Inbound Ports

To display the fixed rate limiting configuration on inbound ports, enter the following command:

```
FastIron#show rate-limit fixed input
Total rate-limited interface count: 11.

 Port    Configured Input Rate        Actual Input Rate
    1                  1000000                  1000000
    3                 10000000                 10005000
    7                 10000000                 10000000
    9                  7500000                  7502000
   11                  8000000                  7999000
   12                  8000000                  7999000
   13                  8000000                  7999000
   14                  8000000                  7999000
   15                  8000000                  7999000
   21                  8000000                  8000000
   25                  7500000                  7502000
```

Syntax: show rate-limit fixed input

The command lists the ports on which fixed rate limiting is configured, and provides the information listed in Table 23.3 for each of the ports.

Table 23.3: CLI Display of Fixed Rate Limiting Information on Inbound Ports

This Field...	Displays...
Total rate-limited interface count	The total number of ports that are configured for Fixed Rate Limiting.
Port	The port number.
Configured Input Rate	The maximum rate requested for inbound traffic. The rate is measured in bits per second (bps).

Table 23.3: CLI Display of Fixed Rate Limiting Information on Inbound Ports (Continued)

This Field...	Displays...
Actual Input Rate	The actual maximum rate provided by the hardware. The rate is measured in bps.

Outbound Ports

To display the fixed rate limiting configuration on outbound ports, enter the following command:

```
FastIron#show rate-limit fixed output
Outbound Rate Shaping Limits in Kbps:

 Port    PortMax   Prio0   Prio1  Prio2  Prio3  Prio4  Prio5  Prio6  Prio7
    1    1000000
    3   10000000
    7   10000000
    9    7500000
   11    8000000
   12    8000000
   13    8000000
   14    8000000
   15    8000000
   21    8000000
   25    7500000
```

Syntax: show rate-limit fixed output

The command lists the ports on which fixed rate limiting is configured, and provides the information listed in Table 23.3 for each of the ports.

Table 23.4: CLI Display of Fixed Rate Limiting Information on Outbound Ports

This Field...	Displays...
Port	The port number.
PortMax	The maximum rate requested for outbound traffic. The rate is measured in bits per second (bps).
Prio0 – Prio7	The port- and priority-based rate limit maximum provided by the hardware. The rate is measured in bps.

Chapter 24
Configuring Traffic Policies

This chapter describes how traffic policies are implemented and configured in the FastIron devices.

About Traffic Policies

FastIron GS, FastIron LS, and FastIron X Series devices use ***traffic policies*** to:

- Rate limit inbound traffic
- Count the packets and bytes per packet to which ACL permit or deny clauses are applied

Traffic policies consist of policy names and policy definitions.

- ***Traffic policy name*** – This is a string of up to 8 alphanumeric characters that identifies individual traffic policy definitions.
- ***Traffic policy definition*** (***TPD***) – This is the command filter associated with a traffic policy name. A TPD can define any one of the following:
    - Rate limiting policy
    - ACL counting policy
    - Combined rate limiting and ACL counting policy

  The maximum number of supported active TPDs is a system-wide parameter and depends on the device you are configuring. The total number of active TPDs cannot exceed the system maximum. See “Maximum Number of Traffic Policies Supported on a Device” on page 24-2.

When you apply a traffic policy to an interface, you do so by adding a reference to the traffic policy in an ACL entry, instead of applying the individual traffic policy to the interface. The traffic policy becomes an ***active traffic policy*** or ***active TPD*** when you bind its associated ACL to an interface.

To configure traffic policies for ACL-based rate limiting, see “Configuring ACL-Based Fixed Rate Limiting” on page 24-4 and “Configuring ACL-Based Adaptive Rate Limiting” on page 24-4.

To configure traffic policies for ACL counting, see “Enabling ACL Counting” on page 24-7.

Configuration Notes and Feature Limitations

Note the following when configuring traffic policies:

- Traffic policies are supported on all FastIron GS and FastIron LS devices, and on FastIron X Series devices running software release 02.3.03 or later.

- This feature is supported in the Layer 2 and Layer 3 code.
- This feature applies to IP ACLs only.
- Traffic policies are not supported on 10-Gigabit Ethernet interfaces.
- The maximum number of supported active TPDs is a system-wide parameter and depends on the device you are configuring. The total number of active TPDs cannot exceed the system maximum. See “Maximum Number of Traffic Policies Supported on a Device” on page 24-2.
- The FastIron GS and FastIron LS devices do not support the use of traffic policies for ACL counting only.
- You can reference the same traffic policy in more than one ACL entry within an access list. For example, two or more ACL statements in ACL 101 can reference a TPD named TPD1.
- You can reference the same traffic policy in more than one access list. For example, ACLs 101 and 102 could both reference a TPD named TPD1.
- To modify or delete an active traffic policy, you must first unbind the ACL that references the traffic policy.
- When you define a TPD (when you enter the CLI command **traffic-policy**), explicit marking of CoS parameters, such as traffic class and 802.1p priority, are not available on the device. In the case of a TPD defining rate limiting, the device re-marks CoS parameters based on the DSCP value in the packet header and the determined conformance level of the rate limited traffic, as shown in Table 24.1.

Table 24.1: CoS Parameters for Packets that use Rate Limiting Traffic Policies

If the packet’s Conformance Level is...	and the packet’s DSCP Value is...	the device sets the Traffic Class and 802.1p Priority to...
0 (Green) or 1 (Yellow)	0 – 7	0 (lowest priority queue)
	8 – 15	1
	16 – 23	2
	24 – 31	3
	32 – 39	4
	40 – 47	5
	48 – 55	6
	56 – 63	7 (highest priority queue)
2 (Red)	N/A	0 (lowest priority queue)

Maximum Number of Traffic Policies Supported on a Device

The maximum number of supported active traffic policies is a system-wide parameter and depends on the device you are configuring, as follows:

- By default, up to 1024 active traffic policies are supported on Layer 2 switches. This value is fixed on Layer 2 switches and cannot be modified.
- The number of active traffic policies supported on Layer 3 switches varies depending on the configuration and the available system memory. The default value and also the maximum number of traffic policies supported on Layer 3 switches is 50.

Setting the Maximum Number of Traffic Policies Supported on a Layer 3 Device

If desired you can adjust the maximum number of active traffic policies that a Layer 3 device will support. To do so, enter commands such as the following at the Global CONFIG level of the CLI:

```
FastIron(config)#system-max hw-traffic-conditioner 25
FastIron(config)#write memory
FastIron(config)#reload
```

NOTE: You must save the configuration and reload the software to place the change into effect.

Syntax: [no] system-max hw-traffic-conditioner <num>

<num> is a value from 0 to *n*, where 0 disables hardware resources for traffic policies, and *n* is a number up to 1024. The maximum number you can configure depends on the configuration and available memory on your device. If the configuration you enter causes the device to exceed the available memory, the device will reject the configuration and display a warning message on the console.

NOTE: Foundry does not recommend setting the system-max for traffic policies to 0 (zero), since this renders traffic policies ineffective.

ACL-Based Rate Limiting via Traffic Policies

Platform Support:

- FESX/FSX/FWSX devices running software release 02.3.03 and later

ACL-based rate limiting provides the facility to limit the rate for IP traffic that matches the permit conditions in extended IP ACLs. This feature is available in the Layer 2 and Layer 3 code.

To configure ACL-based rate limiting on an X Series device, you create individual ***traffic policies***, then reference the traffic policies in one or more ACL entries (also called clauses or statements). The traffic policies become effective on ports to which the ACLs are bound. See "About Traffic Policies" on page 24-1.

When you configure a traffic policy for rate limiting, the device automatically enables ***rate limit counting**,* similar to the two-rate three-color marker (trTCM) mechanism described in RFC 2698 for adaptive rate limiting, and the single-rate three-color marker (srTCM) mechanism described in RFC 2697 for fixed rate limiting. This feature counts the number of bytes and trTCM or srTCM conformance level per packet to which rate limiting traffic policies are applied. See "ACL and Rate Limit Counting" on page 24-7.

You can configure ACL-based rate limiting on the following interface types:

- physical Ethernet interfaces
- virtual interfaces
- trunk ports
- specific VLAN members on a port (New in 02.3.03 – see "Applying an ACL to Specific VLAN Members on a Port (Layer 2 Devices Only)" on page 17-24
- a subset of ports on a virtual interface (New in 02.3.03 – see "Applying an ACL to a Subset of Ports on a Virtual Interface (Layer 3 Devices Only)" on page 17-25.)

Support for Fixed Rate Limiting and Adaptive Rate Limiting

X Series devices support the following types of ACL-based rate limiting:

- **Fixed Rate Limiting** – Enforces a strict bandwidth limit. The device forwards traffic that is within the limit but either drops all traffic that exceeds the limit, or forwards all traffic that exceeds the limit at the lowest priority level, according to the action specified in the traffic policy.
- **Adaptive Rate Limiting** – Enforces a flexible bandwidth limit that allows for bursts above the limit. You can configure Adaptive Rate Limiting to forward, modify the IP precedence of and forward, or drop traffic based on whether the traffic is within the limit or exceeds the limit.

Configuring ACL-Based Fixed Rate Limiting

Use the procedures in this section to configure ACL-based fixed rate limiting. Before configuring this feature, see what to consider in “Configuration Notes and Feature Limitations” on page 24-1.

Fixed rate limiting enforces a strict bandwidth limit. The port forwards traffic that is within the limit. If the port receives more than the specified number of fragments in a one-second interval, the device either drops or forwards subsequent fragments in hardware, depending on the action you specify.

To implement the ACL-based fixed rate limiting feature, first create a traffic policy, then reference the policy in an extended ACL statement. Lastly, bind the ACL to an interface. Follow the steps below.

1. Create a traffic policy. Enter a command such as the following:

   ```
   FastIron(config)#traffic-policy TPD1 rate-limit fixed 100 exceed-action drop
   ```

2. Create an extended ACL entry or modify an existing extended ACL entry that references the traffic policy. For example:

   ```
   FastIron(config)#access-list 101 permit ip host 210.10.12.2 any traffic-policy
   TPD1
   ```

3. Bind the ACL to an interface.

   ```
   FastIron(config)#int e 5
   FastIron(config-if-e5)#ip access-group 101 in
   FastIron(config-if-e5)#exit
   ```

The above commands configure a fixed rate limiting policy that allows port e5 to receive a maximum traffic rate of 100 kbps. If the port receives additional bits during a given one-second interval, the port drops the additional inbound packets that are received within that one-second interval.

Syntax: [no] traffic-policy <TPD name> rate-limit fixed <cir value> exceed-action <action> [count]

Syntax: access-list <num> permit | deny.... traffic policy <TPD name>

Syntax: [no] ip access-group <num> in

NOTES: For brevity, some parameters were omitted from the above **access-list** syntax.

The software allows you to add a reference to a non-existent TPD in an ACL statement and to bind that ACL to an interface. The software does not issue a warning or error message for non-existent TPDs.

Use the **no** form of the command to delete a traffic policy definition. Note that you cannot delete a traffic policy definition if it is currently in use on a port. To delete a traffic policy, first unbind the associated ACL.

<TPD name> is the name of the traffic policy definition. This value can be 8 or fewer alphanumeric characters.

rate-limit fixed specifies that the traffic policy will enforce a strict bandwidth.

<cir value> is the committed information rate in kbps. This value can be from 64 – 1000000 Kbps.

exceed-action <action> specifies the action to be taken when packets exceed the configured cir value. See “Specifying the Action to be Taken for Packets that are Over the Limit” .

The **count** parameter is optional and enables ACL counting. See “ACL and Rate Limit Counting” on page 24-7.

Configuring ACL-Based Adaptive Rate Limiting

Use the procedures in this section to configure ACL-based adaptive rate limiting. Before configuring this feature, see what to consider in “Configuration Notes and Feature Limitations” on page 24-1.

Table 24.2 lists the configurable parameters for ACL-based adaptive rate limiting:

Table 24.2: ACL-Based Adaptive Rate Limiting Parameters

Parameter	Definition
Committed Information Rate (CIR)	The guaranteed kilobit rate of inbound traffic that is allowed on a port.
Committed Burst Size (CBS)	The number of bytes per second allowed in a burst before some packets will exceed the committed information rate. Larger bursts are more likely to exceed the rate limit. The CBS must be a value greater than zero (0). Foundry recommends that this value be equal to or greater than the size of the largest possible IP packet in a stream.
Peak Information Rate (PIR)	The peak maximum kilobit rate for inbound traffic on a port. The PIR must be equal to or greater than the CIR.
Peak Burst Size (PBS)	The number of bytes per second allowed in a burst before all packets will exceed the peak information rate. The PBS must be a value greater than zero (0). Foundry recommends that this value be equal to or greater than the size of the largest possible IP packet in the stream.

If a port receives more than the configured bit or byte rate in a one-second interval, the port will either drop or forward subsequent data in hardware, depending on the action you specify.

To implement the ACL-based adaptive rate limiting feature, first create a traffic policy then reference the policy in an extended ACL statement. Lastly, bind the ACL to an interface. Follow the steps below.

1. Create a traffic policy. Enter a command such as the following:

   ```
   FastIron(config)#traffic-policy TPDAfour rate-limit adaptive cir 10000 cbs 1600
   pir 20000 pbs 4000 exceed-action drop
   ```

2. Create a new extended ACL entry or modify an existing extended ACL entry that references the traffic policy. For example:

   ```
   FastIron(config)#access-list 104 permit ip host 210.10.12.2 any traffic-policy
   TPDAfour
   ```

3. Bind the ACL to an interface.

   ```
   FastIron(config)#int e 7
   FastIron(config-if-e7)#ip access-group 104 in
   FastIron(config-if-e7)#exit
   ```

The above commands configure an adaptive rate limiting policy that enforces a guaranteed committed rate of 10000 kbps on port e7 and allows bursts of up to 1600 bytes. It also enforces a peak rate of 20000 kbps and allows bursts of 4000 bytes above the PIR limit. If the port receives additional bits during a given one-second interval, the port drops all packets on the port until the next one-second interval starts.

Syntax: [no] traffic-policy <TPD name> rate-limit adaptive cir <cir value> cbs <cbs value> pir <pir value> pbs <pbs value> exceed-action <action> [count]

Syntax: access-list <num> permit | deny.... traffic policy <TPD name>

Syntax: [no] ip access-group <num> in

NOTES: For brevity, some parameters were omitted from the above **access-list** syntax.

The software allows you to add a reference to a non-existent TPD in an ACL statement and to bind that ACL to an interface. The software does not issue a warning or error message for non-existent TPDs.

Use the **no** form of the command to delete a traffic policy definition. Note that you cannot delete a traffic policy definition if it is currently in use on a port. To delete a traffic policy, first unbind the associated ACL.

<TPD name> is the name of the traffic policy definition. This value can be 8 or fewer alphanumeric characters.

rate-limit adaptive specifies that the policy will enforce a flexible bandwidth limit that allows for bursts above the limit.

<cir value> is the committed information rate in kbps. See Table 24.2.

<cbs value> is the committed burst size in bytes. See Table 24.2.

<pir value> is the peak information rate in kbps. See Table 24.2.

<pbs value> is the peak burst size in bytes. See Table 24.2.

exceed-action <action> specifies the action to be taken when packets exceed the configured values. See "Specifying the Action to be Taken for Packets that are Over the Limit" .

The **count** parameter is optional and enables ACL counting. See "ACL and Rate Limit Counting" on page 24-7.

Specifying the Action to be Taken for Packets that are Over the Limit

You can specify the action to be taken when packets exceed the configured cir value for fixed rate limiting, or the cir, cbs, pir, and pbs values for adaptive rate limiting. You can specify one of the following actions:

- Drop packets that exceed the limit
- Permit packets that exceed the limit and forward them at the lowest priority level

Dropping Packets that Exceed the Limit

This section shows some example configurations and provides the CLI syntax for configuring a port to drop packets that exceed the configured limits for rate limiting.

EXAMPLES:

The following shows an example fixed rate limiting configuration.

```
FastIron(config)#traffic-policy TPD1 rate-limit fixed 10000 exceed-action drop
```

The above command sets the fragment threshold at 10,000 per second. If the port receives more than 10,000 packet fragments in a one-second interval, the device drops the excess fragments.

Syntax: traffic-policy <TPD name> rate-limit fixed <cir value> exceed-action drop

EXAMPLES:

The following shows an example adaptive rate limiting configuration.

```
FastIron(config)#traffic-policy TPDAfour rate-limit adaptive cir 10000 cbs 1600 pir
20000 pbs 4000 exceed-action drop
```

The above commands configure an adaptive rate limiting policy that enforces a guaranteed committed rate of 10000 kbps on port e7 and allows bursts of up to 1600 bytes. It also enforces a peak rate of 20000 kbps and allows bursts of 4000 bytes above the PIR limit. If the port receives additional bits during a given one-second interval, the port drops all packets on the port until the next one-second interval starts.

Syntax: traffic-policy rate-limit adaptive cir <cir value> cbs <cbs value> pir <pir value> pbs <pbs value> exceed-action drop

Permitting Packets that Exceed the Limit

This section shows some example configurations and provides the CLI syntax for configuring a port to permit packets that exceed the configured limit for rate limiting.

EXAMPLES:

The following shows an example fixed rate limiting configuration.

```
FastIron(config)#traffic-policy TPD1 rate-limit fixed 10000 exceed-action permit-at-
low-pri
```

The above command sets the fragment threshold at 10,000 per second. If the port receives more than 10,000 packet fragments in a one-second interval, the device takes the specified action. The action specified with this command is to permit excess fragments and forward them at the lowest priority level.

Syntax: [no] traffic-policy <TPD name> rate-limit fixed <cir value> exceed-action permit-at-low-pri

EXAMPLES:

The following shows an example adaptive rate limiting configuration.

```
FastIron(config)#traffic-policy TPDAfour rate-limit adaptive cir 10000 cbs 1600 pir
20000 pbs 4000 exceed-action permit-at-low-pri
```

The above commands configure an adaptive rate limiting policy that enforces a guaranteed committed rate of 10000 kbps on port e7 and allows bursts of up to 1600 bytes. It also enforces a peak rate of 20000 kbps and allows bursts of 4000 bytes above the PIR limit. If the port receives additional bits during a given one-second interval, the port permits all packets on the port and forwards the packets at the lowest priority level.

Syntax: traffic-policy rate-limit adaptive cir <cir value> cbs <cbs value> pir <pir value> pbs <pbs value> exceed-action permit-at-low-pri

ACL and Rate Limit Counting

Platform Support:

- FESX/FSX/FWSX devices running software release 02.3.03 and later

ACL counting enables the Foundry device to count the number of packets and the number of bytes per packet to which ACL filters are applied.

Rate limit counting counts the number of bytes and conformance level per packet to which rate limiting traffic policies are applied. The device uses the counting method similar to the two-rate three-color marker (trTCM) mechanism described in RFC 2698 for adaptive rate limiting, and the single-rate three-color marker (srTCM) mechanism described in RFC 2697 for fixed rate limiting. Rate limit counting is automatically enabled when a traffic policy is enforced (active). You can view these counters using the **show** commands listed in “Viewing Traffic Policies” on page 24-10.

For more information about traffic policies, see “About Traffic Policies” on page 24-1.

This section provides the following procedures for ACL counting and rate limit counting:

- “Enabling ACL Counting” on page 24-7
- “Viewing ACL And Rate Limit Counters” on page 24-9
- “Clearing ACL and Rate Limit Counters” on page 24-9

Enabling ACL Counting

NOTE: The FastIron GS and FastIron LS do not support the use of traffic policies for ACL counting only. However, these models do support the use of traffic policies for ACL counting together with rate limiting traffic policies. See “Enabling ACL Counting with Rate Limiting Traffic Policies” on page 24-8.

Use the procedures in this section to configure ACL counting. Before configuring this feature, see what to consider in “Configuration Notes and Feature Limitations” on page 24-1.

To enable ACL counting on an X Series device, first create a ***traffic policy***, then reference the traffic policy in an extended ACL entry. Lastly, bind the ACL to an interface. The ACL counting policy becomes effective on ports to which the ACLs are bound.

You also can enable ACL counting when you create a traffic policy for rate limiting. See “Enabling ACL Counting with Rate Limiting Traffic Policies” on page 24-8.

To implement the ACL counting feature, perform the following steps:

1. Create a traffic policy. Enter a command such as the following:

   ```
   FastIron(config)#traffic-policy TPD5 count
   ```

2. Create an extended ACL entry or modify an existing extended ACL entry that references the traffic policy definition. For example:

   ```
   FastIron(config)#access-list 101 permit ip host 210.10.12.2 any traffic-policy
   TPD5
   ```

3. Bind the ACL to an interface.

   ```
   FastIron(config)#int e 4
   FastIron(config-if-e4)#ip access-group 101 in
   FastIron(config-if-e4)#exit
   ```

The above commands configure an ACL counting policy and apply it to port e4. Port e4 counts the number of packets and the number of bytes on the port that were permitted or denied by ACL filters.

Syntax: [no] traffic-policy <TPD name> count

Syntax: access-list <num> permit | deny.... traffic policy <TPD name>

Syntax: [no] ip access-group <num> in

NOTES: For brevity, some parameters were omitted from the above **access-list** syntax.

The software allows you to add a reference to a non-existent TPD in an ACL statement and to bind that ACL to an interface. The software does not issue a warning or error message for non-existent TPDs.

Use the **no** form of the command to delete a traffic policy definition. Note that you cannot delete a traffic policy definition if it is currently in use on a port. To delete a traffic policy, first unbind the associated ACL.

<TPD name> is the name of the traffic policy definition. This value can be 8 alphanumeric characters or less.

Enabling ACL Counting with Rate Limiting Traffic Policies

The configuration example in the section “Enabling ACL Counting” shows how to enable ACL counting without having to configure parameters for rate limiting. You also can enable ACL counting while defining a rate limiting traffic policy, as illustrated in the following configuration examples.

EXAMPLES:

To enable ACL counting while defining traffic policies for fixed rate limiting, enter commands such as the following at the Global CONFIG Level of the CLI:

```
FastIron(config)#traffic-policy TPD1 rate-limit fixed 1000 count exceed-action drop
FastIron(config)#traffic-policy TPD2 rate-limit fixed 10000 exceed-action drop count
```

Syntax: [no] traffic-policy <TPD name> rate-limit fixed <cir value> exceed-action <action> count

EXAMPLES:

To enable ACL counting while defining traffic policies for adaptive rate limiting, enter commands such as the following at the Global CONFIG Level of the CLI:

```
traffic-policy TPDA4 rate-limit adaptive cir 10000 cbs 1600 pir 20000 pbs 4000 count
exceed-action drop
traffic-policy TPDA5 rate-limit adaptive cir 10000 cbs 1600 pir 20000 pbs 4000
exceed-action permit-at-low-pri count
```

Syntax: traffic-policy rate-limit adaptive cir <cir value> cbs <cbs value> pir <pir value> pbs <pbs value> exceed-action <action> count

Viewing ACL And Rate Limit Counters

When ACL counting is enabled on the Foundry device, you can use **show** commands to display the total packet count and byte count of the traffic filtered by ACL statements. The output of the show commands also display the rate limiting traffic counters, which are automatically enabled for active rate limiting traffic policies.

Use either the **show access-list accounting** command or the **show statistics traffic-policy** command to display ACL and traffic policy counters. The output of these commands are identical. The following shows an example output.

```
FastIron#show access-list accounting g_voip
Traffic Policy - g_voip:
General Counters:
Port Region#                  Byte Count            Packet Count
------------------    --------------------   ----------------------
7 (4/1 - 4/12)                  85367040                    776064
All port regions                84367040                    776064

Rate Limiting Counters:
Port Region#        Green Conformance  Yellow Conformance     Red Conformance
------------------  ------------------  ------------------  ------------------
7 (4/1 - 4/12)      329114195612139520   37533986897781760                   0
All port regions    329114195612139520   37533986897781760                   0
```

Syntax: show access-list accounting traffic-policy [<TPD name>]

or

Syntax: show statistics traffic-policy [<TPD name>]

Table 24.3 explains the output of the **show access-list accounting** and **show statistics traffic-policy** commands.

Table 24.3: ACL and Rate Limit Counting Statistics

This Line...	Displays...
Traffic Policy	The name of the traffic policy.
General Counters:	
Port Region #	The port region to which the active traffic policy applies.
Byte Count	The number of bytes that were filtered (matched ACL clauses).
Packet Count	The number of packets that were filtered (matched ACL clauses).
Rate Limiting Counters:	
Port Region#	The port region to which the active traffic policy applies.
Green Conformance	The number of bytes that did not exceed the CIR packet rate.
Yellow Conformance	The number of bytes that exceeded the CIR packet rate.
Red Conformance	The number of bytes that exceeded the PIR packet rate.

Clearing ACL and Rate Limit Counters

The Foundry device keeps a running tally of the number of packets and the number of bytes per packet that are filtered by ACL statements and rate limiting traffic policies. You can clear these accumulated counters, essentially resetting them to zero. To do so, use either the **clear access-list account traffic-policy** or the **clear statistics traffic-policy** command.

To clear the counters for ACL counting and rate limit counting, enter commands such as the following:

```
FastIron(config)#clear access-list accounting traffic-policy CountOne
FastIron(config)#clear statistics traffic-policy CountTwo
```

Syntax: clear access-list accounting traffic-policy <TPD name>

or

Syntax: clear statistics traffic-policy <TPD name>

where <TPD name> is the name of the traffic policy definition for which you want to clear traffic policy counters.

Viewing Traffic Policies

To view traffic policies that are currently defined on the Foundry device, enter the **show traffic-policy** command. An example display output is shown below. Table 24.4 defines the output.

```
FastIron#show traffic-policy t_voip
Traffic Policy - t_voip:
Metering Enabled, Parameters:
          Mode:  Adaptive Rate-Limiting
          cir: 100 kbps,     cbs: 2000 bytes,     pir: 200 kbps,     pbs: 4000 bytes
Counting Not Enabled
Number of References/Bindings:1
```

Syntax: show traffic-policy [<TPD name>]

To display all traffic policies, enter the **show traffic-policy** command without entering a TPD name.

Table 24.4: Traffic Policy Information

This Line...	Displays...
Traffic Policy	The name of the traffic policy.
Metering	Shows whether or not rate limiting was configured as part of the traffic policy. • Enabled – The traffic policy includes a rate limiting configuration. • Disabled – The traffic policy does not include a rate limiting configuration
Mode	If rate limiting is enabled, this field shows the type of metering enabled on the port: • Fixed Rate-Limiting • Adaptive Rate-Limiting
cir	The committed information rate, in kbps, for the adaptive rate-limiting policy.
cbs	The committed burst size, in bytes per second, for the adaptive rate-limiting policy.
pir	The peak information rate, in kbps, for the adaptive rate-limiting policy.
pbs	The peak burst size, in bytes per second, for the adaptive rate-limiting policy.
Counting	Shows whether or not ACL counting was configured as part of the traffic policy. • Enabled – Traffic policy includes an ACL counting configuration. • Disabled – Traffic policy does not include an ACL traffic counting configuration.

Table 24.4: Traffic Policy Information (Continued)

This Line...	Displays...
Number of References/Bindings	The number of times this traffic policy is referenced in an ACL statement and the number of active bindings for this traffic policy.

Chapter 25

Configuring LLDP and LLDP-MED

LLDP and LLDP-MED are supported in the following releases:

- FESX/FSX/FWSX devices running software release 04.0.00 or later – L2, BL3, L3
- FGS and FLS devices running software release 04.0.00 or later – L2, BL3

This chapter describes how to configure the following protocols:

Link Layer Discovery Protocol (LLDP) – The Layer 2 network discovery protocol described in the IEEE 802.1AB standard, *Station and Media Access Control Connectivity Discovery*. This protocol enables a station to advertise its capabilities to, and to discover, other LLDP-enabled stations in the same 802 LAN segments.

LLDP Media Endpoint Devices (LLDP-MED) – The Layer 2 network discovery protocol extension described in the ANSI/TIA-1057 standard, *LLDP for Media Endpoint Devices*. This protocol enables a switch to configure and manage connected Media Endpoint devices that need to send media streams across the network (e.g., IP telephones and security cameras).

LLDP enables network discovery between Network Connectivity devices (such as switches), whereas LLDP-MED enables network discovery at the edge of the network, between Network Connectivity devices and media Endpoint devices (such as IP phones).

The information generated via LLDP and LLDP-MED can be used to diagnose and troubleshoot misconfigurations on both sides of a link. For example, the information generated can be used to discover devices with misconfigured or unreachable IP addresses, and to detect port speed and duplex mismatches.

LLDP and LLDP-MED facilitate interoperability across multiple vendor devices. Foundry devices running LLDP can interoperate with third-party devices running LLDP.

Foundry's LLDP and LLDP-MED implementation adheres to the IEEE 802.1AB and TIA-1057 standards.

Terms Used in this Chapter

Endpoint Device – An LLDP-MED device located at the network edge, that provides some aspect of IP communications service based on IEEE 802 LAN technology. An Endpoint device is classified in one of three class types (I, II, or III) and can be an IP telephone, softphone, VoIP gateway, or conference bridge, among others.

LLDP Agent – The protocol entity that implements LLDP for a particular IEEE 802 device. Depending on the configured LLDP operating mode, an LLDP agent can send and receive LLDP advertisements (frames), or send LLDP advertisements only, or receive LLDP advertisements only.

LLDPDU (LLDP Data Unit) – A unit of information in an LLDP packet that consists of a sequence of short variable length information elements, known as **TLVs.**

MIB (Management Information Base) – A virtual database that identifies each manageable object by its name, syntax, accessibility, and status, along with a text description and unique object identifier (OID). The database is accessible by a Network Management Station (NMS) using a management protocol such as the Simple Network Management Protocol (SNMP).

Network Connectivity Device – A forwarding 802 LAN device, such as a router, switch, or wireless access point.

Station – A node in a network.

TLV (Type-Length-Value) – An information element in an LLDPDU that describes the *type* of information being sent, the *length* of the information string, and the *value* (actual information) that will be transmitted.

TTL (Time-to-Live) – Specifies the length of time that the receiving device should maintain the information acquired via LLDP in its MIB.

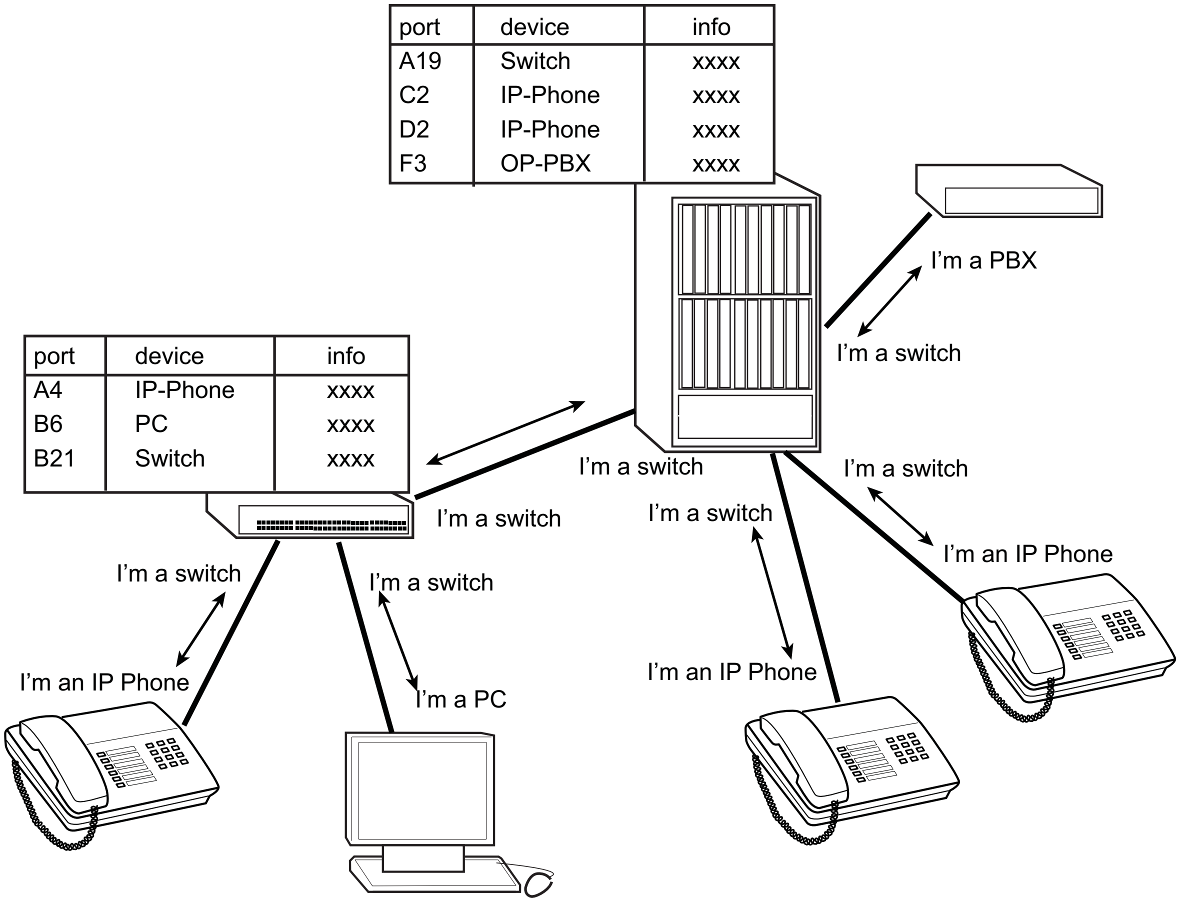
LLDP Overview

LLDP enables a station attached to an IEEE 802 LAN/MAN to advertise its capabilities to, and to discover, other stations in the same 802 LAN segments.

The information distributed via LLDP (the advertisement) is stored by the receiving device in a standard Management Information Base (MIB), accessible by a Network Management System (NMS) using a management protocol such as the Simple Network Management Protocol (SNMP). The information also can be viewed via the CLI, using **show LLDP** commands.

Figure 25.1 illustrates LLDP connectivity.

Figure 25.1 LLDP Connectivity



port	device	info
A19	Switch	xxxx
C2	IP-Phone	xxxx
D2	IP-Phone	xxxx
F3	OP-PBX	xxxx

port	device	info
A4	IP-Phone	xxxx
B6	PC	xxxx
B21	Switch	xxxx

Benefits of LLDP

LLDP provides the following benefits:

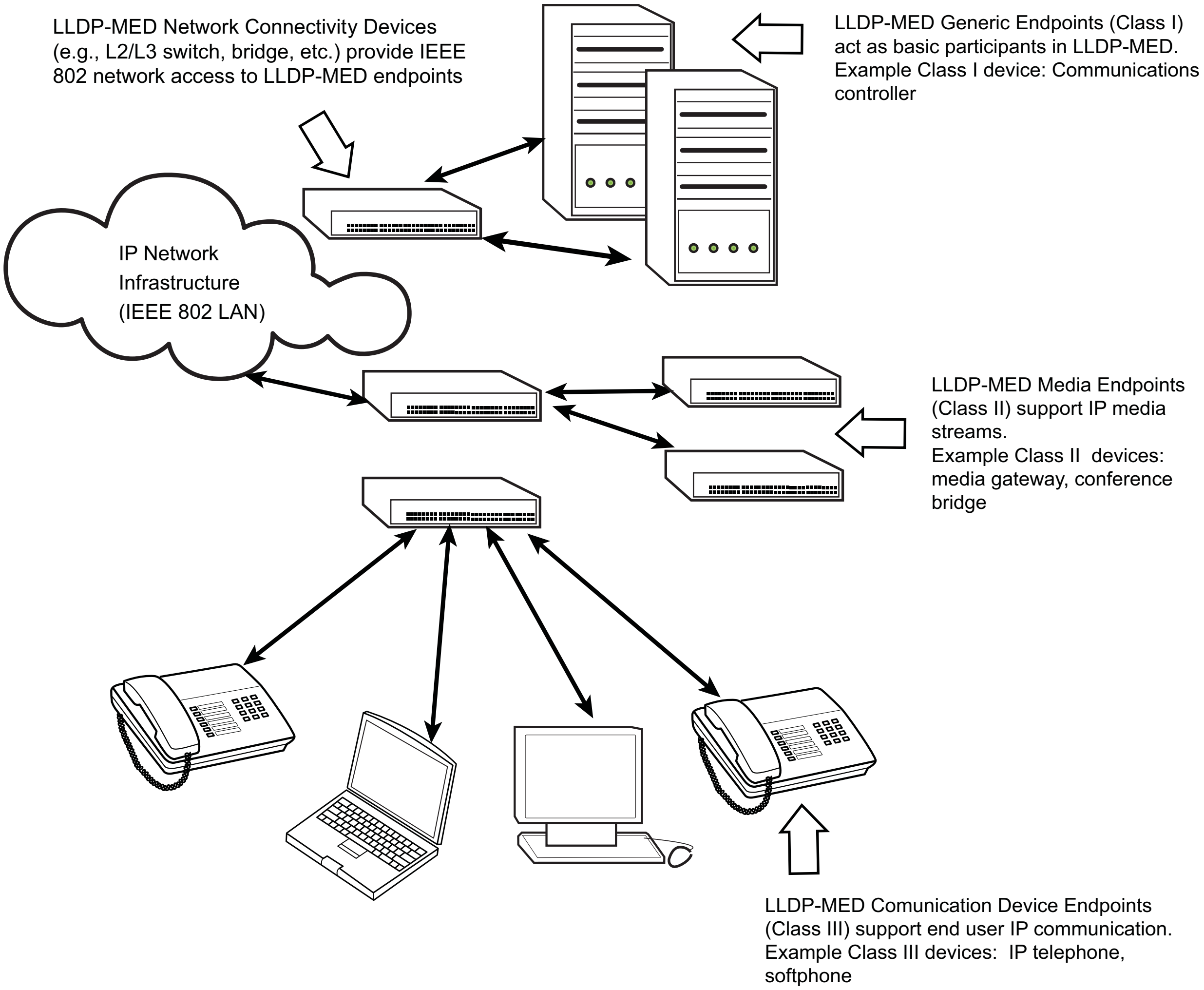
- Network Management
    - Simplifies the use of and enhances the ability of network management tools in multi-vendor environments
    - Enables discovery of accurate physical network topologies such as which devices are neighbors and through which ports they connect
    - Enables discovery of stations in multi-vendor environments
- Network Inventory Data
    - Supports optional system name, system description, system capabilities and management address
    - System description can contain the device's product name or model number, version of hardware type, and operating system
    - Provides device capability, such as switch, router, or WLAN access port
- Network troubleshooting
    - Information generated via LLDP can be used to detect speed and duplex mismatches
    - Accurate topologies simplify troubleshooting within enterprise networks
    - Can discover devices with misconfigured or unreachable IP addresses

LLDP-MED Overview

LLDP-MED is an extension to LLDP. This protocol enables advanced LLDP features in a Voice over IP (VoIP) network. Whereas LLDP enables network discovery between Network Connectivity devices, LLDP-MED enables network discovery between Network Connectivity devices and media Endpoints such as, IP telephones, softphones, VoIP gateways and conference bridges.

Figure 25.2 demonstrates LLDP-MED connectivity.

Figure 25.2 LLDP-MED Connectivity


LLDP-MED Network Connectivity Devices (e.g., L2/L3 switch, bridge, etc.) provide IEEE 802 network access to LLDP-MED endpoints
LLDP-MED Generic Endpoints (Class I) act as basic participants in LLDP-MED. Example Class I device: Communications controller
IP Network Infrastructure (IEEE 802 LAN)
LLDP-MED Media Endpoints (Class II) support IP media streams. Example Class II devices: media gateway, conference bridge
LLDP-MED Comunication Device Endpoints (Class III) support end user IP communication. Example Class III devices: IP telephone, softphone

Benefits of LLDP-MED

LLDP-MED provides the following benefits:

- Vendor-independent management capabilities, enabling different IP telephony systems to interoperate in one network.
- Automatically deploys network policies, such as Layer 2 and Layer 3 QoS policies and Voice VLANs.
- Supports E-911 Emergency Call Services (ECS) for IP telephony
- Collects Endpoint inventory information
- Network troubleshooting
    - Helps to detect improper network policy configuration

LLDP-MED Class

An LLDP-MED class specifies an Endpoint's type and its capabilities. An Endpoint can belong to one of three LLDP-MED class types:

- **Class 1 (Generic Endpoint)** – A Class 1 Endpoint requires basic LLDP discovery services, but does not support IP media nor does it act as an end-user communication appliance. A Class 1 Endpoint can be an IP communications controller, other communication-related server, or other device requiring basic LLDP discovery services.

- **Class 2 (Media Endpoint)** – A Class 2 Endpoint supports media streams and may or may not be associated with a particular end user. Device capabilities include media streaming, as well as all of the capabilities defined for Class 1 Endpoints. A Class 2 Endpoint can be a voice/media gateway, conference, bridge, media server, etc..
- **Class 3 (Communication Endpoint)** – A Class 3 Endpoint supports end user IP communication. Capabilities include aspects related to end user devices, as well as all of the capabilities defined for Class 1 and Class 2 Endpoints. A Class 3 Endpoint can be an IP telephone, softphone (PC-based phone), or other communication device that directly supports the end user.

  Discovery services defined in Class 3 include location identifier (ECS/E911) information and inventory management.

The LLDP-MED device class is advertised when LLDP-MED is enabled on a port.

Figure 25.2 illustrates LLDP-MED connectivity and supported LLDP-MED classes.

General Operating Principles

LLDP and LLDP-MED use the services of the Data Link sublayers, Logical Link Control and Media Access Control, to transmit and receive information to and from other ***LLDP Agents*** (protocol entities that implement LLDP).

LLDP is a one-way protocol. An LLDP agent can transmit and receive information to and from another LLDP agent located on an adjacent device, but it cannot solicit information from another LLDP agent, nor can it acknowledge information received from another LLDP agent.

Operating Modes

When LLDP is enabled on a global basis, by default, each port on the Foundry device will be capable of transmitting and receiving LLDP packets. You can disable a port’s ability to transmit and receive LLDP packets, or change the operating mode to one of the following:

- Transmit LLDP information only
- Receive LLDP information only

Transmit Mode

An LLDP agent sends LLDP packets to adjacent LLDP-enabled devices. The LLDP packets contain information about the transmitting device and port.

An LLDP agent initiates the transmission of LLDP packets whenever the transmit countdown timing counter expires, or whenever LLDP information has changed. When a transmit cycle is initiated, the LLDP manager extracts the MIB objects and formats this information into TLVs. The TLVs are inserted into an LLDPDU, addressing parameters are prepended to the LLDPDU, and the information is sent out LLDP-enabled ports to adjacent LLDP-enabled devices.

Receive Mode

An LLDP agent receives LLDP packets from adjacent LLDP-enabled devices. The LLDP packets contain information about the transmitting device and port.

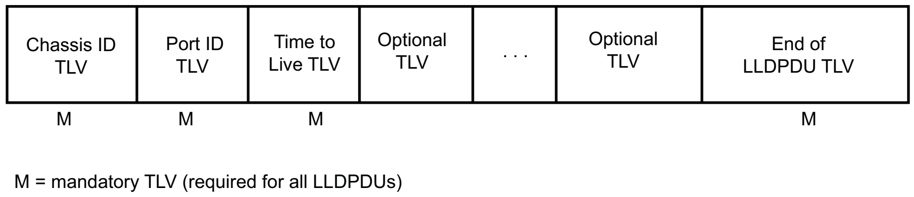
When an LLDP agent receives LLDP packets, it checks to ensure that the LLDPDUs contain the correct sequence of mandatory TLVs, then validates optional TLVs. If the LLDP agent detects any errors in the LLDPDUs and TLVs, it drops them in software. TLVs that are not recognized but do not contain basic formatting errors, are assumed to be valid and are assigned a temporary identification index and stored for future possible alter retrieval by network management. All validated TLVs are stored in the neighbor database.

LLDP Packets

LLDP agents transmit information about a sending device/port in packets called LLDP Data Units (LLDPDUs). All the LLDP information to be communicated by a device is contained within a single 1500 byte packet. A device receiving LLDP packets is not permitted to combine information from multiple packets.

As shown in Figure 25.3, each LLDPDU has three mandatory TLVs, an End of LLDPDU TLV, plus optional TLVs as selected by network management.

Figure 25.3 LLDPDU Packet Format



Each LLDPDU consists of an untagged Ethernet header and a sequence of short, variable length information elements known as TLVs.

TLVs have *Type*, *Length*, and *Value* fields, where:

- ***Type*** identifies the kind of information being sent
- ***Length*** indicates the length (in octets) of the information string
- ***Value*** is the actual information being sent (for example, a binary bit map or an alpha-numeric string containing one or more fields).

TLV Support

This section lists the LLDP and LLDP-MED TLV support.

LLDP TLVs

There are two types of LLDP TLVs, as specified in the IEEE 802.3AB standard:

- ***Basic Management TLVs*** consist of both *optional* general system information TLVs as well as *mandatory* TLVs.

  Mandatory TLVs cannot be manually configured. They are always the first three TLVs in the LLDPDU, and are part of the packet header.

  General system information TLVs are optional in LLDP implementations and are defined by the Network Administrator.

  Foundry devices support the following Basic Management TLVs:

  - Chassis ID (mandatory)
  - Port ID (mandatory)
  - Time to Live (mandatory)
  - Port description
  - System name
  - System description
  - System capabilities
  - Management address
  - End of LLDPDU

- **Organizationally-specific TLVs** are optional in LLDP implementations and are defined and encoded by individual organizations or vendors. These TLVs include support for, but are not limited to, the IEEE 802.1 and 802.3 standards and the TIA-1057 standard.

  Foundry devices support the following Organizationally-specific TLVs:

  - **802.1 organizationally-specific TLVs**

    Port VLAN ID

    VLAN name TLV

  - **802.3 organizationally-specific TLVs**

    MAC/PHY configuration/status

    Power via MDI

    Link aggregation

    Maximum frame size

LLDP-MED TLVs

Foundry devices honor and send the following LLDP-MED TLVs, as defined in the TIA-1057 standard:

- LLDP-MED capabilities
- Network policy
- Location identification
- Extended power-via-MDI

Mandatory TLVs

When an LLDP agent transmits LLDP packets to other agents in the same 802 LAN segments, the following mandatory TLVs are always included:

- Chassis ID
- Port ID
- Time to Live (TTL)

This section describes the above TLVs in detail.

Chassis ID

The Chassis ID identifies the device that sent the LLDP packets.

There are several ways in which a device may be identified. A chassis ID subtype, included in the TLV and shown in Table 25.1, indicates how the device is being referenced in the Chassis ID field.

Table 25.1: Chassis ID Subtypes

ID Subtype	Description
0	Reserved
1	Chassis component
2	Interface alias
3	Port component
4	MAC address
5	Network address
6	Interface name
7	Locally assigned
8 – 255	Reserved

Foundry devices use chassis ID subtype 4, the base MAC address of the device. Other third party devices may use a chassis ID subtype other than 4. The chassis ID will appear similar to the following on the remote device, and in the CLI display output on the Foundry device (**show lldp local-info**):

```
Chassis ID (MAC address):  0012.f233.e2c0
```

The chassis ID TLV is always the first TLV in the LLDPDU.

Port ID

The Port ID identifies the port from which LLDP packets were sent.

There are several ways in which a port may be identified, as shown in Table 25.2. A port ID subtype, included in the TLV, indicates how the port is being referenced in the Port ID field.

Table 25.2: Port ID Subtypes

ID Subtype	Description
0	Reserved
1	Interface alias
2	Port component
3	MAC address
4	Network address
5	Interface name
6	Agent circuit ID
7	Locally assigned
8 – 255	Reserved

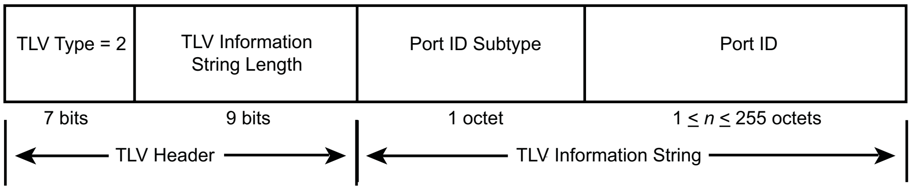
Foundry devices use port ID subtype 3, the permanent MAC address associated with the port. Other third party devices may use a port ID subtype other than 3. The port ID appears similar to the following on the remote device, and in the CLI display output on the Foundry device (show lldp local-info):

```
Port ID (MAC address):  0012.f233.e2d3
```

The LLDPDU format is shown in “ LLDPDU Packet Format” on page 25-6.

The Port ID TLV format is shown below.

Figure 25.4 Port ID TLV Packet Format



TTL Value

The Time to Live (TTL) Value is the length of time the receiving device should maintain the information acquired via LLDP in its MIB.

The TTL value is automatically computed based on the LLDP configuration settings. The TTL value will appear similar to the following on the remote device, and in the CLI display output on the Foundry device (show lldp local-info):

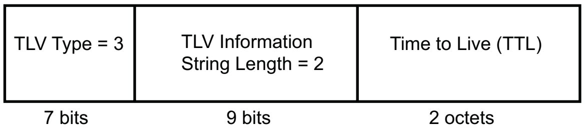
```
Time to live: 40 seconds
```

- If the TTL field has a value other than zero, the receiving LLDP agent is notified to completely replace all information associated with the LLDP agent/port with the information in the received LLDPDU.
- If the TTL field value is zero, the receiving LLDP agent is notified that all system information associated with the LLDP agent/port is to be deleted. This TLV may be used, for example, to signal that the sending port has initiated a port shutdown procedure.

The LLDPDU format is shown in “ LLDPDU Packet Format” on page 25-6.

The TTL TLV format is shown below.

Figure 25.5 TTL TLV Packet Format



MIB Support

Foundry devices support the following standard MIB modules:

- LLDP-MIB
- LLDP-EXT-DOT1-MIB
- LLDP-EXT-DOT3-MIB
- LLDP-EXT-MED-MIB

Syslog Messages

Syslog messages for LLDP provide management applications with information related to MIB data consistency and general status. These Syslog messages correspond to the lldpRemTablesChange SNMP notifications. See "Enabling LLDP SNMP Notifications and Syslog Messages" on page 25-13.

Syslog messages for LLDP-MED provide management applications with information related to topology changes. These Syslog messages correspond to the lldpXMedTopologyChangeDetected SNMP notifications. See "Enabling SNMP Notifications and Syslog Messages for LLDP-MED Topology Changes" on page 25-24.

Configuring LLDP

This section describes how to enable and configure LLDP.

Table 25.3 lists the LLDP global-level tasks and the default behavior/value for each task.

Table 25.3: LLDP Global Configuration Tasks and Default Behavior / Value

Global Task	Default Behavior / Value when LLDP is enabled
Enabling LLDP on a global basis	Disabled
Specifying the maximum number of LLDP neighbors per device	Automatically set to 392 neighbors per device
Specifying the maximum number of LLDP neighbors per port	Automatically set to 4 neighbors per port
Enabling SNMP notifications and Syslog messages	Disabled
Changing the minimum time between SNMP traps and Syslog messages	Automatically set to 2 seconds when SNMP notifications and Syslog messages for LLDP are enabled
Enabling and disabling TLV advertisements	When LLDP transmit is enabled, by default, the Foundry device will automatically advertise LLDP capabilities, except for the system description, VLAN name, and power-via-MDI information, which may be configured by the system administrator. Also, if desired, you can disable the advertisement of individual TLVs.
Changing the minimum time between LLDP transmissions	Automatically set to 2 seconds
Changing the interval between regular LLDP transmissions	Automatically set to 30 seconds
Changing the holdtime multiplier for transmit TTL	Automatically set to 4
Changing the minimum time between port reinitializations	Automatically set to 2 seconds

Configuration Notes and Considerations

- LLDP is supported on Ethernet interfaces only.
- If a port is 802.1X-enabled, the transmission and reception of LLDP packets will only take place while the port

is authorized.

- Cisco Discovery Protocol (CDP) and Foundry Discovery Protocol (FDP) run independently of LLDP. Therefore, these discovery protocols can run simultaneously on the same device.
- By default, the Foundry device limits the number of neighbors per port to four, and staggers the transmission of LLDP packets on different ports, in order to minimize any high-usage spikes to the CPU.
- Ports that are in blocking mode (spanning tree) can still receive LLDP packets from a forwarding port.
- Auto-negotiation status indicates what is being advertised by the port for 802.3 auto-negotiation.

Enabling and Disabling LLDP

LLDP is enabled by default on individual ports. However, to run LLDP, you must first enable it on a global basis (on the entire device).

To enable LLDP globally, enter the following command at the global CONFIG level of the CLI:

```
FastIron(config)#lldp run
```

Syntax: [no] lldp run

Use the [no] form of the command to disable LLDP.

Changing a Port's LLDP Operating Mode

LLDP packets are not exchanged until LLDP is enabled on a global basis. When LLDP is enabled on a global basis, by default, each port on the Foundry device will be capable of transmitting and receiving LLDP packets. You can disable a port's ability to transmit and receive LLDP packets, or change the operating mode to one of the following:

- Transmit LLDP information only
- Receive LLDP information only

You can configure a different operating mode for each port on the Foundry device. For example, you could disable the receipt and transmission of LLDP packets on port e 2/1, configure port e 2/3 to only receive LLDP packets, and configure port e 2/5 to only transmit LLDP packets.

The following sections show how to change the operating mode.

Enabling and Disabling Receive and Transmit Mode

To disable the receipt and transmission of LLDP packets on individual ports, enter a command such as the following at the Global CONFIG level of the CLI:

```
FastIron(config)#no lldp enable ports e 2/4 e 2/5
```

The above command disables LLDP on ports 2/4 and 2/5. These ports will not transmit nor receive LLDP packets.

To enable LLDP on a port after it has been disabled, enter the following command:

```
FastIron(config)#lldp enable ports e 2/4
```

Syntax: [no] lldp enable ports ethernet <port-list> | all

Use the [no] form of the command to disable the receipt and transmission of LLDP packets on a port.

For <port-list>, specify the port(s) in one of the following formats:

- FastIron GS and LS compact switches – <stacknum/slotnum/portnum>
- FastIron chassis devices – <slotnum/portnum>
- FESX, and FWSX compact switches – <portnum>

You can list all of the ports individually, use the keyword **to** to specify ranges of ports, or a combination of both. To apply the configuration to all ports on the device, use the keyword **all** instead of listing the ports individually.

NOTE: When a port is configured to both receive and transmit LLDP packets and the MED capabilities TLV is enabled, LLDP-MED is enabled as well. LLDP-MED is not enabled if the operating mode is set to *receive only* or *transmit only*.

Enabling and Disabling Receive Only Mode

When LLDP is enabled on a global basis, by default, each port on the Foundry device will be capable of transmitting and receiving LLDP packets. To change the LLDP operating mode from *receive and transmit* mode to *receive only* mode, simply disable the transmit mode. Enter a command such as the following at the Global CONFIG level of the CLI:

```
FastIron(config)#no lldp enable transmit ports e 2/4 e 2/5 e 2/6
```

The above command changes the LLDP operating mode on ports 2/4, 2/5, and 2/6 from *transmit and receive* mode to *receive only* mode.

To change a port's LLDP operating mode from *transmit only* to *receive only*, first disable the *transmit only* mode, then enable the *receive only* mode. Enter commands such as the following:

```
FastIron(config)#no lldp enable transmit ports e 2/7 e 2/8 e 2/9
FastIron(config)#lldp enable receive ports e 2/7 e 2/8 e 2/9
```

The above commands change the LLDP operating mode on ports 2/7, 2/8, and 2/9, from *transmit only* to *receive only*. Note that if you do not disable the *transmit only* mode, you will configure the port to both transmit and receive LLDP packets.

NOTE: LLDP-MED is not enabled when you enable the *receive only* operating mode. To enable LLDP-MED, you must configure the port to both receive and transmit LLDP packets. See "Enabling and Disabling Receive and Transmit Mode" on page 25-11.

Syntax: [no] lldp enable receive ports ethernet <port-list> | all

Use the [no] form of the command to disable the *receive only* mode.

For <port-list>, specify the port(s) in one of the following formats:

- FastIron GS and LS compact switches – <stacknum/slotnum/portnum>
- FastIron chassis devices – <slotnum/portnum>
- FESX, and FWSX compact switches – <portnum>

You can list all of the ports individually, use the keyword **to** to specify ranges of ports, or a combination of both. To apply the configuration to all ports on the device, use the keyword **all** instead of listing the ports individually.

Enabling and Disabling Transmit Only Mode

When LLDP is enabled on a global basis, by default, each port on the Foundry device will be capable of transmitting and receiving LLDP packets. To change the LLDP operating mode to *transmit only* mode, simply disable the *receive* mode. Enter a command such as the following at the Global CONFIG level of the CLI:

```
FastIron(config)#no lldp enable receive ports e 2/4 e 2/5 e 2/6
```

The above command changes the LLDP operating mode on ports 2/4, 2/5, and 2/6 from *transmit and receive* mode to *transmit only* mode. Any incoming LLDP packets will be dropped in software.

To change a port's LLDP operating mode from *receive only* to *transmit only*, first disable the *receive only* mode, then enable the *transmit only* mode. For example, enter commands such as the following at the Global CONFIG level of the CLI:

```
FastIron(config)#no lldp enable receive ports e 2/7 e 2/8
FastIron(config)#lldp enable transmit ports e 2/7 e 2/8
```

The above commands change the LLDP operating mode on ports 2/7 and 2/8 from *receive only* mode to *transmit only* mode. Any incoming LLDP packets will be dropped in software. Note that if you do not disable *receive only* mode, you will configure the port to both receive and transmit LLDP packets.

NOTE: LLDP-MED is not enabled when you enable the *transmit only* operating mode. To enable LLDP-MED, you must configure the port to both receive and transmit LLDP packets. See "Enabling and Disabling Receive and Transmit Mode" on page 25-11.

Syntax: [no] lldp enable transmit ports ethernet <port-list> | all

Use the [no] form of the command to disable the *transmit only* mode.

For <port-list>, specify the port(s) in one of the following formats:

- FastIron GS and LS compact switches – <stacknum/slotnum/portnum>
- FastIron chassis devices – <slotnum/portnum>
- FESX, and FWSX compact switches – <portnum>

You can list all of the ports individually, use the keyword **to** to specify ranges of ports, or a combination of both. To apply the configuration to all ports on the device, use the keyword **all** instead of listing the ports individually.

Specifying the Maximum Number of LLDP Neighbors

You can change the limit of the number of LLDP neighbors for which LLDP data will be retained, per device as well as per port.

Per Device

You can change the maximum number of neighbors for which LLDP data will be retained for the entire system.

For example, to change the maximum number of LLDP neighbors for the entire device to 26, enter the following command:

```
FastIron(config)#lldp max-total-neighbors 26
```

Syntax: [no] lldp max-total-neighbors <value>

Use the [no] form of the command to remove the static configuration and revert to the default value of 392.

where <value> is a number between 16 and 65536. The default number of LLDP neighbors per device is 392.

Use the **show lldp** command to view the configuration.

Per Port

You can change the maximum number of LLDP neighbors for which LLDP data will be retained for each port. By default, the maximum number is four and you can change this to a value between one and 64.

For example, to change the maximum number of LLDP neighbors to six, enter the following command:

```
FastIron(config)#lldp max-neighbors-per-port 6
```

Syntax: [no] lldp max-neighbors-per-port <value>

Use the [no] form of the command to remove the static configuration and revert to the default value of four.

where <value> is a number from 1 to 64. The default is number of LLDP neighbors per port is four.

Use the **show lldp** command to view the configuration.

Enabling LLDP SNMP Notifications and Syslog Messages

SNMP notifications and Syslog messages for LLDP provide management applications with information related to MIB data updates and general status.

When you enable LLDP SNMP notifications, corresponding Syslog messages are enabled as well. When you enable LLDP SNMP notifications, the device will send traps and corresponding Syslog messages whenever there are changes to the LLDP data received from neighboring devices.

LLDP SNMP notifications and corresponding Syslog messages are disabled by default. To enable them, enter a command such as the following at the Global CONFIG level of the CLI:

```
FastIron(config)#lldp enable snmp notifications ports e 4/2 to 4/6
```

The above command enables SNMP notifications and corresponding Syslog messages on ports 4/2 and 4/6. By default, the device will send no more than one SNMP notification and Syslog message within a five second period. If desired, you can change this interval. See “Specifying the Minimum Time Between SNMP Traps and Syslog Messages” on page 25-14.

Syntax: [no] lldp enable snmp notifications ports ethernet <port-list> | all

For <port-list>, specify the port(s) in one of the following formats:

- FastIron GS and LS compact switches – <stacknum/slotnum/portnum>
- FastIron chassis devices – <slotnum/portnum>
- FESX, and FWSX compact switches – <portnum>

You can list all of the ports individually, use the keyword **to** to specify ranges of ports, or a combination of both. To apply the configuration to all ports on the device, use the keyword **all** instead of listing the ports individually.

Specifying the Minimum Time Between SNMP Traps and Syslog Messages

When SNMP notifications and Syslog messages for LLDP are enabled, the device will send no more than one SNMP notification and corresponding Syslog message within a five second period. If desired, you can throttle the amount of time between transmission of SNMP traps (lldpRemTablesChange) and Syslog messages from five seconds up to a value equal to one hour (3600 seconds).

NOTE: Because LLDP Syslog messages are rate limited, some LLDP information given by the system will not match the current LLDP statistics (as shown in the **show lldp statistics** command output).

To change the minimum time interval between traps and Syslog messages, enter a command such as the following:

```
FastIron(config)#lldp snmp-notification-interval 60
```

When the above command is applied, the LLDP agent will send no more than one SNMP notification and Syslog message every 60 seconds.

Syntax: [no] lldp snmp-notification-interval <seconds>

where <seconds> is a value between 5 and 3600. The default is 5 seconds.

Changing the Minimum Time between LLDP Transmissions

The LLDP transmit delay timer limits the number of LLDP frames an LLDP agent can send within a specified time frame. When you enable LLDP, the system automatically sets the LLDP transmit delay timer to two seconds. If desired, you can change the default behavior from two seconds to a value between 1 and 8192 seconds.

NOTE: The LLDP transmit delay timer must not be greater than one quarter of the LLDP transmission interval (CLI command **lldp transmit-interval**).

The LLDP transmit delay timer prevents an LLDP agent from transmitting a series of successive LLDP frames during a short time period, when rapid changes occur in LLDP. It also increases the probability that multiple changes, rather than single changes, will be reported in each LLDP frame.

To change the LLDP transmit delay timer, enter a command such as the following at the Global CONFIG level of the CLI:

```
FastIron(config)#lldp transmit-delay 7
```

The above command causes the LLDP agent to wait a minimum of seven seconds after transmitting an LLDP frame and before sending another LLDP frame.

Syntax: [no] lldp transmit-delay <seconds>.

where <seconds> is a value between 1 and 8192. The default is two seconds. Note that this value must not be greater than one quarter of the LLDP transmission interval (CLI command **lldp transmit-interval**).

Changing the Interval between Regular LLDP Transmissions

The LLDP transmit interval specifies the number of seconds between regular LLDP packet transmissions. When you enable LLDP, by default, the device will wait 30 seconds between regular LLDP packet transmissions. If desired, you can change the default behavior from 30 seconds to a value between 5 and 32768 seconds.

To change the LLDP transmission interval, enter a command such as the following at the Global CONFIG level of the CLI:

```
FastIron(config)#lldp transmit-interval 40
```

The above command causes the LLDP agent to transmit LLDP frames every 40 seconds.

Syntax: [no] lldp transmit-interval <seconds>

where <seconds> is a value from 5 to 32768. The default is 30 seconds.

NOTE: Setting the transmit interval and/or transmit holdtime multiplier to inappropriate values can cause the LLDP agent to transmit LLDPDUs with TTL values that are excessively high. This in turn can affect how long a receiving device will retain the information if it is not refreshed.

Changing the HoldTime Multiplier for Transmit TTL

The holdtime multiplier for transmit TTL is used to compute the actual time-to-live (TTL) value used in an LLDP frame. The TTL value is the length of time the receiving device should maintain the information in its MIB. When you enable LLDP, the device automatically sets the holdtime multiplier for TTL to four. If desired, you can change the default behavior from four to a value between two and ten.

To compute the TTL value, the system multiplies the LLDP transmit interval by the holdtime multiplier. For example, if the LLDP transmit interval is 30 and the holdtime multiplier for TTL is 4, then the value 120 is encoded in the TTL field in the LLDP header.

To change the holdtime multiplier, enter a command such as the following at the Global CONFIG level of the CLI:

```
FastIron(config)#lldp transmit-hold 6
```

Syntax: [no] lldp transmit-hold <value>.

where <value> is a number from 2 to 10. The default value is 4.

NOTE: Setting the transmit interval and/or transmit holdtime multiplier to inappropriate values can cause the LLDP agent to transmit LLDPDUs with TTL values that are excessively high. This in turn can affect how long a receiving device will retain the information if it is not refreshed.

Changing the Minimum Time between Port Reinitializations

The LLDP re-initialization delay timer specifies the minimum number of seconds the device will wait from when LLDP is disabled on a port, until it will honor a request to re-enable LLDP on that port. When you enable LLDP, the system sets the re-initialization delay timer to two seconds. If desired, you can change the default behavior from two seconds to a value between one and ten seconds.

To set the re-initialization delay timer, enter a command such as the following at the Global CONFIG level of the CLI:

```
FastIron(config)#lldp reinit-delay 5
```

The above command causes the device to wait five seconds after LLDP is disabled, before attempting to honor a request to re-enable it.

Syntax: [no] lldp reinit-delay <seconds>

where <seconds> is a value from 1 – 10. The default is two seconds.

LLDP TLVs Advertised by the Foundry Device

When LLDP is enabled on a global basis, the Foundry device will automatically advertise the following information, except for the features noted:

General system information:

- Management address
- Port description
- System capabilities
- System description (not automatically advertised)
- System name

802.1 capabilities:

- VLAN name (not automatically advertised)
- Untagged VLAN ID

802.3 capabilities:

- Link aggregation information
- MAC/PHY configuration and status
- Maximum frame size
- Power-via-MDI information (not automatically advertised)

The above TLVs are described in detail in the following sections.

NOTE: The system description, VLAN name, and power-via-MDI information TLVs are not automatically enabled. The following sections show how to enable these advertisements.

General System Information

Except for the system description, the Foundry device will advertise the following system information when LLDP is enabled on a global basis:

- Management address
- Port description
- System capabilities
- System description (not automatically advertised)

- System name

Management Address

The management address is an IPv4 address that can be used to manage the device. If no management address is explicitly configured to be advertised, the Foundry device will use the first available IPv4 address configured on the following types of interfaces, in the following order of preference:

- Physical port on which LLDP will be transmitting the packet
- Loopback interface
- Virtual routing interface (VE)
- Router interface on a VLAN that the port is a member of
- Other physical interface

If no IP address is configured, the port's current MAC address will be advertised.

The management address will appear similar to the following on the remote device, and in the CLI display output on the Foundry device (**show lldp local-info**):

```
Management address (IPv4): 209.157.2.1
```

Port Description

The port description TLV identifies the port from which the LLDP agent transmitted the advertisement. The port description is taken from the ifDescr MIB object from MIB-II.

By default, the port description is automatically advertised when LLDP is enabled on a global basis. To disable advertisement of the port description, enter a command such as the following:

```
FastIron(config)#no lldp advertise port-description ports e 2/4 to 2/12
```

The port description will appear similar to the following on the remote device, and in the CLI display output on the Foundry device (**show lldp local-info**):

```
Port description: “GigabitEthernet20”
```

Syntax: [no] lldp advertise port-description ports ethernet <port-list> | all

For <port-list>, specify the port(s) in one of the following formats:

- FastIron GS and LS compact switches – <stacknum/slotnum/portnum>
- FastIron chassis devices – <slotnum/portnum>
- FESX, and FWSX compact switches – <portnum>

You can list all of the ports individually, use the keyword **to** to specify ranges of ports, or a combination of both. To apply the configuration to all ports on the device, use the keyword **all** instead of listing the ports individually. Note that using the keyword **all** may cause undesirable effects on some ports. For example, if you configure **all** ports to advertise their VLAN name, and the configuration includes ports that are not members of any VLAN, the system will warn of the misconfigurations on non-member VLAN ports. The configuration will be applied to all ports, however, the ports that are not members of any VLAN will not send VLAN name advertisements.

System Capabilities

The system capabilities TLV identifies the primary function(s) of the device and indicates whether these primary functions are enabled. The primary functions can be one or more of the following (more than one for example, if the device is both a bridge and a router):

- Repeater
- Bridge
- WLAN access point
- Router
- Telephone

- DOCSIS cable device
- Station only (devices that implement end station capability)
- Other

System capabilities for Foundry devices are based on the type of software image in use (e.g., Layer 2 switch or Layer 3 router). The enabled capabilities will be the same as the available capabilities, except that when using a router image (base or full Layer 3), if the global route-only feature is turned on, the bridge capability will not be included, since no bridging takes place.

By default, the system capabilities are automatically advertised when LLDP is enabled on a global basis. To disable this advertisement, enter a command such as the following:

```
FastIron(config)#no lldp advertise system-capabilities ports e 2/4 to 2/12
```

The system capabilities will appear similar to the following on the remote device, and in the CLI display output on the Foundry device (**show lldp local-info**):

```
    System capabilities :   bridge
    Enabled capabilities:   bridge
```

Syntax: [no] lldp advertise system-capabilities ports ethernet <port-list> | all

For <port-list>, specify the port(s) in one of the following formats:

- FastIron GS and LS compact switches – <stacknum/slotnum/portnum>
- FastIron chassis devices – <slotnum/portnum>
- FESX, and FWSX compact switches – <portnum>

You can list all of the ports individually, use the keyword **to** to specify ranges of ports, or a combination of both. To apply the configuration to all ports on the device, use the keyword **all** instead of listing the ports individually. Note that using the keyword **all** may cause undesirable effects on some ports. For example, if you configure **all** ports to advertise their VLAN name, and the configuration includes ports that are not members of any VLAN, the system will warn of the misconfigurations on non-member VLAN ports. The configuration will be applied to all ports, however, the ports that are not members of any VLAN will not send VLAN name advertisements.

System Description

The system description is the network entity, which can include information such as the product name or model number, the version of the system's hardware type, the software operating system level, and the networking software version. The information corresponds to the sysDescr MIB object in MIB-II.

To advertise the system description, enter a command such as the following:

```
FastIron(config)#lldp advertise system-description ports e 2/4 to 2/12
```

The system description will appear similar to the following on the remote device, and in the CLI display output on the Foundry device (**show lldp local-info**):

```
    + System description  : "Foundry Networks, Inc. FESX424-PREM-PoE, IronWare V\
                             ersion 04.0.00b256T3e1 Compiled on Sep 04 2007 at 0\
                             3:54:29 labeled as SXS04000b256"
```

Syntax: [no] lldp advertise system-description ports ethernet <port-list> | all

For <port-list>, specify the port(s) in one of the following formats:

- FastIron GS and LS compact switches – <stacknum/slotnum/portnum>
- FastIron chassis devices – <slotnum/portnum>
- FESX, and FWSX compact switches – <portnum>

You can list all of the ports individually, use the keyword **to** to specify ranges of ports, or a combination of both. To apply the configuration to all ports on the device, use the keyword **all** instead of listing the ports individually. Note that using the keyword **all** may cause undesirable effects on some ports. For example, if you configure **all** ports to advertise their VLAN name, and the configuration includes ports that are not members of any VLAN, the system will warn of the misconfigurations on non-member VLAN ports. The configuration will be applied to all ports, however, the ports that are not members of any VLAN will not send VLAN name advertisements.

System Name

The system name is the system's administratively assigned name, taken from the sysName MIB object in MIB-II. The sysName MIB object corresponds to the name defined with the CLI command **hostname**.

By default, the system name is automatically advertised when LLDP is enabled on a global basis. To disable this advertisement, enter a command such as the following:

```
FastIron(config)#no lldp advertise system-name ports e 2/4 to 2/12
```

The system name will appear similar to the following on the remote device, and in the CLI display output on the Foundry device (**show lldp local-info**):

```
    System name:  “FESX424_POE”
```

Syntax: [no] lldp advertise system-name ports ethernet <port-list> | all

For <port-list>, specify the port(s) in one of the following formats:

- FastIron GS and LS compact switches – <stacknum/slotnum/portnum>
- FastIron chassis devices – <slotnum/portnum>
- FESX, and FWSX compact switches – <portnum>

You can list all of the ports individually, use the keyword **to** to specify ranges of ports, or a combination of both. To apply the configuration to all ports on the device, use the keyword **all** instead of listing the ports individually. Note that using the keyword **all** may cause undesirable effects on some ports. For example, if you configure **all** ports to advertise their VLAN name, and the configuration includes ports that are not members of any VLAN, the system will warn of the misconfigurations on non-member VLAN ports. The configuration will be applied to all ports, however, the ports that are not members of any VLAN will not send VLAN name advertisements.

802.1 Capabilities

Except for the VLAN name, the Foundry device will advertise the following 802.1 attributes when LLDP is enabled on a global basis:

- VLAN name (not automatically advertised)
- Untagged VLAN ID

VLAN Name

The VLAN name TLV contains the name and VLAN ID of a VLAN configured on a port. An LLDPDU may include multiple instances of this TLV, each for a different VLAN.

To advertise the VLAN name, enter a command such as the following:

```
FastIron(config)#lldp advertise vlan-name vlan 99 ports e 2/4 to 2/12
```

The VLAN name will appear similar to the following on the remote device, and in the CLI display output on the Foundry device (**show lldp local-info**):

```
    VLAN name (VLAN 99): “Voice-VLAN-99”
```

Syntax: [no] lldp advertise vlan-name vlan <vlan ID> ports ethernet <port-list> | all

For <vlan ID>, enter the VLAN ID to advertise.

For <port-list>, specify the port(s) in one of the following formats:

- FastIron GS and LS compact switches – <stacknum/slotnum/portnum>
- FastIron chassis devices – <slotnum/portnum>
- FESX, and FWSX compact switches – <portnum>

You can list all of the ports individually, use the keyword **to** to specify ranges of ports, or a combination of both. To apply the configuration to all ports on the device, use the keyword **all** instead of listing the ports individually. Note that using the keyword **all** may cause undesirable effects on some ports. For example, if you configure **all** ports to advertise their VLAN name, and the configuration includes ports that are not members of any VLAN, the system will warn of the misconfigurations on non-member VLAN ports. The configuration will be applied to all ports, however, the ports that are not members of any VLAN will not send VLAN name advertisements.

Untagged VLAN ID

The port VLAN ID TLV advertises the Port VLAN Identifier (PVID) that will be associated with untagged or priority-tagged frames. If the port is not an untagged member of any VLAN (i.e., the port is strictly a tagged port), the value zero will indicate that.

By default, the port VLAN ID is automatically advertised when LLDP is enabled on a global basis. To disable this advertisement, enter a command such as the following:

```
FastIron(config)#no lldp advertise port-vlan-id ports e 2/4 to 2/12
```

The untagged VLAN ID will appear similar to the following on the remote device, and in the CLI display output on the Foundry device (**show lldp local-info**):

```
    Port VLAN ID: 99
```

Syntax: [no] lldp advertise port-vlan-id ports ethernet <port-list> | all

For <port-list>, specify the port(s) in one of the following formats:

- FastIron GS and LS compact switches – <stacknum/slotnum/portnum>
- FastIron chassis devices – <slotnum/portnum>
- FESX, and FWSX compact switches – <portnum>

You can list all of the ports individually, use the keyword **to** to specify ranges of ports, or a combination of both. To apply the configuration to all ports on the device, use the keyword **all** instead of listing the ports individually. Note that using the keyword **all** may cause undesirable effects on some ports. For example, if you configure **all** ports to advertise their VLAN name, and the configuration includes ports that are not members of any VLAN, the system will warn of the misconfigurations on non-member VLAN ports. The configuration will be applied to all ports, however, the ports that are not members of any VLAN will not send VLAN name advertisements.

802.3 Capabilities

Except for Power-via-MDI information, the Foundry device will advertise the following 802.3 attributes when LLDP is enabled on a global basis:

- Link aggregation information
- MAC/PHY configuration and status
- Maximum frame size
- Power-via-MDI information (not automatically advertised)

Link Aggregation

The **link-aggregation** TLV indicates the following:

- Whether the link is capable of being aggregated
- Whether the link is currently aggregated
- The primary trunk port

Foundry devices advertise link aggregation information about standard link aggregation (LACP) as well as static trunk configuration.

By default, link-aggregation information is automatically advertised when LLDP is enabled on a global basis. To disable this advertisement, enter a command such as the following:

```
FastIron(config)#no lldp advertise link-aggregation ports e 2/12
```

Syntax: [no] lldp advertise link-aggregation ports ethernet <port-list> | all

The link aggregation advertisement will appear similar to the following on the remote device, and in the CLI display output on the Foundry device (**show lldp local-info**):

```
    Link aggregation: not capable
```

For <port-list>, specify the port(s) in one of the following formats:

- FastIron GS and LS compact switches – <stacknum/slotnum/portnum>
- FastIron chassis devices – <slotnum/portnum>
- FESX, and FWSX compact switches – <portnum>

You can list all of the ports individually, use the keyword **to** to specify ranges of ports, or a combination of both. To apply the configuration to all ports on the device, use the keyword **all** instead of listing the ports individually. Note that using the keyword **all** may cause undesirable effects on some ports. For example, if you configure **all** ports to advertise their VLAN name, and the configuration includes ports that are not members of any VLAN, the system will warn of the misconfigurations on non-member VLAN ports. The configuration will be applied to all ports, however, the ports that are not members of any VLAN will not send VLAN name advertisements.

MAC/PHY Configuration Status

The MAC/PHY configuration and status TLV includes the following information:

- Auto-negotiation capability and status
- Speed and duplex mode
- Flow control capabilities for auto-negotiation
- Port speed down-shift and maximum port speed advertisement
- If applicable, indicates if the above settings are the result of auto-negotiation during link initiation or of a manual set override action

The advertisement reflects the effects of the following CLI commands:

- speed-duplex
- flow-control
- gig-default
- link-config

By default, the MAC/PHY configuration and status information are automatically advertised when LLDP is enabled on a global basis. To disable this advertisement, enter a command such as the following:

```
FastIron(config)#no lldp advertise mac-phy-config-status ports e 2/4 to 2/12
```

The MAC/PHY configuration advertisement will appear similar to the following on the remote device, and in the CLI display output on the Foundry device (**show lldp local-info**):

```
    + 802.3 MAC/PHY       : auto-negotiation enabled
      Advertised capabilities: 10baseT-HD, 10baseT-FD, 100baseTX-HD, 100baseTX-FD,
      fdxSPause, fdxBPause, 1000baseT-HD, 1000baseT-FD
      Operational MAU type: 100BaseTX-FD
```

Syntax: [no] lldp advertise mac-phy-config-status ports ethernet <port-list> | all

For <port-list>, specify the port(s) in one of the following formats:

- FastIron GS and LS compact switches – <stacknum/slotnum/portnum>
- FastIron chassis devices – <slotnum/portnum>
- FESX, and FWSX compact switches – <portnum>

You can list all of the ports individually, use the keyword **to** to specify ranges of ports, or a combination of both. To apply the configuration to all ports on the device, use the keyword **all** instead of listing the ports individually. Note that using the keyword **all** may cause undesirable effects on some ports. For example, if you configure **all** ports to advertise their VLAN name, and the configuration includes ports that are not members of any VLAN, the system will warn of the misconfigurations on non-member VLAN ports. The configuration will be applied to all ports, however, the ports that are not members of any VLAN will not send VLAN name advertisements.

Maximum Frame Size

The maximum frame size TLV provides the maximum 802.3 frame size capability of the port. This value is expressed in octets and includes the four-octet Frame Check Sequence (FCS). The default maximum frame size is 1522. The advertised value may change depending on whether the **aggregated-vlan** or **jumbo** CLI commands are in effect.

By default, the maximum frame size is automatically advertised when LLDP is enabled on a global basis. To disable this advertisement, enter a command such as the following:

```
FastIron(config)#no lldp advertise max-frame-size ports e 2/4 to 2/12
```

The maximum frame size advertisement will appear similar to the following on the remote device, and in the CLI display output on the Foundry device (**show lldp local-info**):

```
    Maximum frame size: 1522 octets
```

Syntax: [no] lldp advertise max-frame-size ports ethernet <port-list> | all

For <port-list>, specify the port(s) in one of the following formats:

- FastIron GS and LS compact switches – <stacknum/slotnum/portnum>
- FastIron chassis devices – <slotnum/portnum>
- FESX, and FWSX compact switches – <portnum>

You can list all of the ports individually, use the keyword **to** to specify ranges of ports, or a combination of both. To apply the configuration to all ports on the device, use the keyword **all** instead of listing the ports individually. Note that using the keyword **all** may cause undesirable effects on some ports. For example, if you configure **all** ports to advertise their VLAN name, and the configuration includes ports that are not members of any VLAN, the system will warn of the misconfigurations on non-member VLAN ports. The configuration will be applied to all ports, however, the ports that are not members of any VLAN will not send VLAN name advertisements.

Power-via-MDI

The power-via-MDI TLV provides general information about Power over Ethernet (POE) capabilities and status of the port. It indicates the following:

- POE capability (supported or not supported)
- POE status (enabled or disabled)
- Power Sourcing Equipment (PSE) power pair – indicates which pair of wires is in use and whether the pair selection can be controlled. Foundry's implementation always uses pair A, and cannot be controlled.
- Power class – Indicates the range of power that the connected powered device has negotiated or requested.

NOTE: The power-via-MDI TLV described in this section applies to LLDP. There is also a power-via-MDI TLV for LLDP-MED devices, which provides extensive POE information. See "Extended Power-via-MDI Information" on page 25-34.

To advertise the power-via-MDI information, enter a command such as the following:

```
FastIron(config)#lldp advertise power-via-mdi ports e 2/4 to 2/12
```

The power-via-MDI advertisement will appear similar to the following on the remote device, and in the CLI display output on the Foundry device (**show lldp local-info**):

```
+ 802.3 Power via MDI: PSE port, power enabled, class 0
  Power Pair         : A (not controllable)
```

Syntax: [no] lldp advertise power-via-mdi ports ethernet <port-list> | all

For <port-list>, specify the port(s) in one of the following formats:

- FastIron GS and LS compact switches – <stacknum/slotnum/portnum>
- FastIron chassis devices – <slotnum/portnum>
- FESX, and FWSX compact switches – <portnum>

You can list all of the ports individually, use the keyword **to** to specify ranges of ports, or a combination of both. To apply the configuration to all ports on the device, use the keyword **all** instead of listing the ports individually. Note that using the keyword **all** may cause undesirable effects on some ports. For example, if you configure **all** ports to advertise their VLAN name, and the configuration includes ports that are not members of any VLAN, the system will warn of the misconfigurations on non-member VLAN ports. The configuration will be applied to all ports, however, the ports that are not members of any VLAN will not send VLAN name advertisements.

Configuring LLDP-MED

This section provides the details for configuring LLDP-MED.

Table 25.4 lists the global and interface-level tasks and the default behavior/value for each task.

Table 25.4: LLDP-MED Configuration Tasks and Default Behavior / Value

Task	Default Behavior / Value
Global CONFIG-Level Tasks	
Enabling LLDP-MED on a global basis	Disabled
Enabling SNMP notifications and Syslog messages for LLDP-MED topology change	Disabled
Changing the Fast Start Repeat Count	The system automatically sets the fast start repeat count to 3 when a Network Connectivity Device receives an LLDP packet from an Endpoint that is newly connected to the network. **NOTE:** The LLDP-MED fast start mechanism is only intended to run on links between Network Connectivity devices and Endpoint devices. It does not apply to links between LAN infrastructure elements, including between Network Connectivity devices, or to other types of links.
Interface-Level Tasks	
Defining a location ID	Not configured
Defining a network policy	Not configured

Enabling LLDP-MED

When LLDP is enabled globally, LLDP-MED is enabled if the LLDP-MED capabilities TLV is also enabled. By default, the LLDP-MED capabilities TLV is automatically enabled. To enable LLDP, see “Enabling and Disabling LLDP” on page 25-11.

NOTE: LLDP-MED is not enabled on ports where the LLDP operating mode is *receive only* or *transmit only.* LLDP-MED is enabled on ports that are configured to both receive and transmit LLDP packets and have the LLDP-MED capabilities TLV enabled.

Enabling SNMP Notifications and Syslog Messages for LLDP-MED Topology Changes

SNMP notifications and Syslog messages for LLDP-MED provide management applications with information related to topology changes. For example, SNMP notifications can alert the system whenever a remote Endpoint device is connected to or removed from a local port. SNMP notifications identify the local port where the topology change occurred, as well as the device capability of the remote Endpoint device that was connected to or removed from the port.

When you enable LLDP-MED SNMP notifications, corresponding Syslog messages are enabled as well. When you enable LLDP-MED SNMP notifications, the device will send traps and Syslog messages when an LLDP-MED Endpoint’s neighbor entry is added or removed.

SNMP notifications and corresponding Syslog messages are disabled by default. To enable them, enter a command such as the following at the Global CONFIG level of the CLI:

```
FastIron(config)#lldp enable snmp med-topo-change-notifications ports e 4/4 to 4/6
```

Syntax: [no] lldp enable snmp med-topo-change-notifications ports ethernet <port-list> | all

For <port-list>, specify the port(s) in one of the following formats:

- FastIron GS and LS compact switches – <stacknum/slotnum/portnum>
- FastIron chassis devices – <slotnum/portnum>
- FESX, and FWSX compact switches – <portnum>

You can list all of the ports individually, use the keyword **to** to specify ranges of ports, or a combination of both. To apply the configuration to all ports on the device, use the keyword **all** instead of listing the ports individually. Note that using the keyword **all** may cause undesirable effects on some ports. For example, if you configure **all** ports to advertise their VLAN name, and the configuration includes ports that are not members of any VLAN, the system will warn of the misconfigurations on non-member VLAN ports. The configuration will be applied to all ports, however, the ports that are not members of any VLAN will not send VLAN name advertisements.

Changing the Fast Start Repeat Count

The fast start feature enables a Network Connectivity Device to initially advertise itself at a faster rate for a limited time when an LLDP-MED Endpoint has been newly detected/connected to the network. This feature is important within a VoIP network, for example, where rapid availability is crucial for applications such as emergency call service location (E911).

The fast start timer starts when a Network Connectivity Device receives the first LLDP frame from a newly detected Endpoint.

The ***LLDP-MED fast start repeat count*** specifies the number of LLDP packets that will be sent during the LLDP-MED fast start period. By default, the device will send three packets at one-second intervals. If desired, you can change the number of packets the device will send per second, up to a maximum of 10.

NOTE: The LLDP-MED fast start mechanism is only intended to run on links between Network Connectivity devices and Endpoint devices. It does not apply to links between LAN infrastructure elements, including between Network Connectivity devices, or to other types of links.

To change the LLDP-MED fast start repeat count, enter commands such as the following:

```
FastIron(config)#lldp med fast-start-repeat-count 5
```

The above command causes the device to send five LLDP packets during the LLDP-MED fast start period.

Syntax: [no] lldp med fast-start-repeat-count <value>

where value is a number from 1 to 10, which specifies the number of packets that will be sent during the LLDP-MED fast start period. The default is 3.

Defining a Location ID

The LLDP-MED Location Identification extension enables the Foundry device to set the physical location that an attached Class III Endpoint will use for location-based applications. This feature is important for applications such as IP telephony, for example, where emergency responders need to quickly determine the physical location of a user in North America that has just dialed 911.

For each port, you can define one or more of the following location ID formats:

- Geographic location (coordinate-based)
- Civic address
- Emergency Call Services (ECS) Emergency Location Identification Number (ELIN)

The above location ID formats are defined in the following sections.

Coordinate-Based Location

Coordinate-based location is based on the IETF RFC 3825 [6] standard, which specifies a Dynamic Host Configuration Protocol (DHCP) option for the coordinate-based geographic location of a client.

When you configure an Endpoint's location information using the coordinate-based location, you specify the latitude, longitude, and altitude, along with ***resolution indicators*** (a measure of the accuracy of the coordinates), and the reference ***datum*** (the map used for the given coordinates).

To configure a coordinate-based location for an Endpoint device, enter a command such as the following at the Global CONFIG level of the CLI:

```
FastIron(config)#lldp med location-id coordinate-based latitude
-78.303 resolution 20 longitude 34.27 resolution 18 altitude meters 50 resolution 16
wgs84
```

Syntax: [no] lldp med location-id coordinate-based
latitude <degrees> resolution <bits>
longitude <degrees> resolution <bits>
altitude floors <number> resolution <bits> | meters <number> resolution <bits>
<datum>

latitude <degrees> is the angular distance north or south from the earth's equator measured through 90 degrees. Positive numbers indicate a location north of the equator and negative numbers indicate a location south of the equator.

resolution <bits> specifies the precision of the value given for latitude. A smaller value increases the area within which the device is located. For latitude, enter a number between 1 and 34.

longitude <degrees> is the angular distance from the intersection of the zero meridian. Positive values indicate a location east of the prime meridian and negative numbers indicate a location west of the prime meridian.

resolution <bits> specifies the precision of the value given for longitude. A smaller value increases the area within which the device is located. For longitude resolution, enter a number between 1 and 34.

altitude floors <number> is the vertical elevation of a building above the ground, where 0 represents the floor level associated with the ground level at the main entrance and larger values represent floors that are above (higher in altitude) floors with lower values. For example, 2 for the 2nd floor. Sub-floors can be represented by non-integer values. For example, a mezzanine between floor 1 and floor 2 could be represented as 1.1. Similarly, the mezzanines between floor 4 and floor 5 could be represented as 4.1 and 4.2 respectively. Floors located below ground level could be represented by negative values.

resolution <bits> specifies the precision of the value given for altitude. A smaller value increases the area within which the device is located. For floors resolution, enter the value 0 if the floor is unknown, or 30 if a valid floor is being specified.

altitude meters <number> is the vertical elevation in number of meters, as opposed to floors.

resolution <bits> specifies the precision of the value given for altitude. A smaller value increases the area within which the device is located. For meters resolution, enter a value from 0 to 30.

<Datum> is the map used as the basis for calculating the location. Specify one of the following:

- **wgs84** – (geographical 3D) – World Geodesic System 1984, CRS Code 4327, Prime Meridian Name: Greenwich
- **nad83-navd88** – North American Datum 1983, CRS Code 4269, Prime Meridian Name: Greenwich; The associated vertical datum is the North American Vertical Datum of 1988 (NAVD88). Use this datum when referencing locations on land. If land is near tidal water, use nad83-mllw (below).
- **nad83-mllw** – North American Datum 1983, CRS Code 4269, Prime Meridian Name: Greenwich; The associated vertical datum is mean lower low water (MLLW). Use this datum when referencing locations on water, sea, or ocean.

Example Coordinate-based Location Configuration

The following shows an example coordinate-based location configuration for the Sears Tower, at the following location:

103rd Floor
233 South Wacker Drive
Chicago, IL 60606

```
FastIron(config)#lldp med location-id coordinate-based latitude 41.87884
resolution 18 longitude 87.63602 resolution 18 altitude floors 103 resolution 30
wgs84
```

The above configuration shows the following:

- Latitude is 41.87884 degrees north (or 41.87884 degrees).
- Longitude is 87.63602 degrees west (or 87.63602 degrees).
- The latitude and longitude resolution of 18 describes a geo-location area that is latitude 41.8769531 to latitude 41.8789062 and extends from -87.6367188 to -87.6347657 degrees longitude. This is an area of approximately 373412 square feet (713.3 ft. x 523.5 ft.).
- The location is inside a structure, on the 103rd floor.
- The WGS 84 map was used as the basis for calculating the location.

Example Coordinate-based Location Advertisement

The coordinate-based location advertisement will appear similar to the following on the remote device, and in the CLI display output on the Foundry device (**show lldp local-info**):

```
+ MED Location ID
  Data Format: Coordinate-based
  Latitude Resolution  : 20 bits
  Latitude Value       : -78.303 degrees
  Longitude Resolution : 18 bits
  Longitude Value      : 34.27 degrees
  Altitude Resolution  : 16 bits
  Altitude Value       : 50. meters
  Datum                : WGS 84
```

Civic Address Location

When you configure a media Endpoint's location using the address-based location, you specify the location the entry refers to, the country code, and the elements that describe the civic or postal address.

To configure a civic address-based location for LLDP-MED, enter commands such as the following at the Global CONFIG level of the CLI:

```
FastIron(config)#lldp med location-id civic-address refers-to client country US elem
1 CA elem 3 “Santa Clara” elem 6 “4980 Great America Pkwy” elem 24 95054 elem 27 5
elem 28 551 elem 29 office elem 23 “John Doe”
```

Syntax: [no] lldp med location-id civic-address refers-to <elem> country <country code> elem <CA type> <value> [elem <CA type> <value>] [elem <CA type> <value>]....

refers-to <elem> describes the location that the entry refers to. Specify one of the following:

- client
- dhcp-server
- network-element

where **dhcp-server** or **network-element** should only be used if it is known that the Endpoint is in close physical proximity to the DHCP server or network element.

<country code> is the two-letter ISO 3166 country code in capital ASCII letters. For example:

- CA – Canada
- DE – Germany
- JP – Japan
- KR – Korea
- US – United States

<CA type> is a value from 0 – 255, that describes the civic address element. For example, a CA type of 24 specifies a postal/zip code. Valid elements and their types are listed in Table 25.5.

<value> is the actual value of the elem <CA type>, above. For example, 95123 for the postal/zip code. Acceptable values are listed in Table 25.5, below.

NOTE: If the value of an element contains one or more spaces, use double quotation marks (“) at the beginning and end of the string. For example, `elem 3 “Santa Clara”`.

Table 25.5: Elements used with Civic Address

Civic Address (CA) Type	Description	Acceptable Values / Examples
0	Language	The ISO 639 language code used for presenting the address information.
1	National subdivisions (state, canton, region, province, or prefecture)	Examples: Canada – Province Germany – State Japan – Metropolis Korea – Province United States – State
2	County, parish, gun (JP), or district (IN)	Examples: Canada – County Germany – County Japan – City or rural area Korea – County United States – County
3	City, township, or shi (JP)	Examples: Canada – City or town Germany – City Japan – Ward or village Korea – City or village United States – City or town

Table 25.5: Elements used with Civic Address (Continued)

Civic Address (CA) Type	Description	Acceptable Values / Examples
4	City division, borough, city district, ward, or chou (JP)	Examples: Canada – N/A Germany – District Japan – Town Korea – Urban district United States – N/A
5	Neighborhood or block	Examples: Canada – N/A Germany – N/A Japan – City district Korea – Neighborhood United States – N/A
6	Street	Examples: Canada – Street Germany – Street Japan – Block Korea – Street United States – Street
16	Leading street direction	N (north), E (east), S (south), W (west), NE, NW, SE, SW
17	Trailing street suffix	N (north), E (east), S (south), W (west), NE, NW, SE, SW
18	Street suffix	Acceptable values for the United States are listed in the United States Postal Service Publication 28 [18], Appendix C. Example: Ave, Place
19	House number	The house number (street address) Example: 1234
20	House number suffix	A modifier to the house number. It does not include parts of the house number. Example: A, 1/2
21	Landmark or vanity address	A string name for a location. It conveys a common local designation of a structure, a group of buildings, or a place that helps to locate the place. Example: UC Berkeley
22	Additional location information	An unstructured string name that conveys additional information about the location. Example: west wing

Table 25.5: Elements used with Civic Address (Continued)

Civic Address (CA) Type	Description	Acceptable Values / Examples
23	Name (residence and office occupant)	Identifies the person or organization associated with the address. Example: Textures Beauty Salon
24	Postal / zip code	The valid postal / zip code for the address. Example: 95054-1234
25	Building (structure)	The name of a single building if the street address includes more than one building or if the building name is helpful in identifying the location. Example: Law Library
26	Unit (apartment, suite)	The name or number of a part of a structure where there are separate administrative units, owners, or tenants, such as separate companies or families who occupy that structure. Common examples include suite or apartment designations. Example: Apt 27
27	Floor	Example: 4
28	Room number	The smallest identifiable subdivision of a structure. Example: 7A
29	Placetype	The type of place described by the civic coordinates. For example, a home, office, street, or other public space. Example: Office
30	Postal community name	When the postal community name is defined, the civic community name (typically CA type 3) is replaced by this value. Example: Alviso
31	Post office box (P.O. box)	When a P.O. box is defined, the street address components (CA types 6, 16, 17, 18, 19, and 20) are replaced with this value. Example: P.O. Box 1234
32	Additional code	An additional country-specific code that identifies the location. For example, for Japan, this is the Japan Industry Standard (JIS) address code. The JIS address code provides a unique address inside of Japan, down to the level of indicating the floor of the building.
128	Script	The *script* (from ISO 15924 [14]) used to present the address information. Example: Latn **NOTE:** If not manually configured, the system assigns the default value **Latn**
255	Reserved	

Example Civic Address Location Advertisement

The Civic address location advertisement will appear similar to the following on the remote device, and in the CLI display output on the Foundry device (**show lldp local-info**):

```
+ MED Location ID
  Data Format: Civic Address
  Location of: Client
  Country    : "US"
  CA Type    : 1
  CA Value   : "CA"
  CA Type    : 3
  CA Value   : "Santa Clara"
  CA Type    : 6
  CA Value   : "4980 Great America Pkwy."
  CA Type    : 24
  CA Value   : "95054"
  CA Type    : 27
  CA Value   : "5"
  CA Type    : 28
  CA Value   : "551"
  CA Type    : 29
  CA Value   : "office"
  CA Type    : 23
  CA Value   : "John Doe"
```

Emergency Call Services

The Emergency Call Service (ECS) location is used specifically for Emergency Call Services applications.

When you configure a media Endpoint's location using the emergency call services location, you specify the Emergency Location Identification Number (ELIN) from the North America Numbering Plan format, supplied to the Public Safety Answering Point (PSAP) for ECS purposes.

To configure an ECS-based location for LLDP-MED, enter a command such as the following at the Global CONFIG level of the CLI:

```
FastIron(config)#lldp med location-id ecs-elin 4082071700
```

Syntax: [no] lldp med location-id ecs-elin <number> ports ethernet <port-list> | all

<number> is a number from 10 to 25 digits in length.

For <port-list>, specify the port(s) in one of the following formats:

- FastIron GS and LS compact switches – <stacknum/slotnum/portnum>
- FastIron chassis devices – <slotnum/portnum>
- FESX, and FWSX compact switches – <portnum>

You can list all of the ports individually, use the keyword **to** to specify ranges of ports, or a combination of both. To apply the configuration to all ports on the device, use the keyword **all** instead of listing the ports individually.

Example ECS ELIN Location Advertisements

The ECS ELIN location advertisement will appear similar to the following on the remote device, and in the CLI display output on the Foundry device (**show lldp local-info**):

```
+ MED Location ID
  Data Format: ECS ELIN
  Value      : 4082071700
```

Defining an LLDP-MED Network Policy

An LLDP-MED network policy defines an Endpoint's VLAN configuration (VLAN type and VLAN ID) and associated Layer 2 and Layer 3 priorities that apply to a specific set of applications on a port.

NOTE: This feature applies to applications that have specific real-time network policy requirements, such as interactive voice or video services. It is not intended to run on links other than between Network Connectivity devices and Endpoints, and therefore does not advertise the multitude of network policies that frequently run on an aggregated link.

To define an LLDP-MED network policy for an Endpoint, enter a command such as the following:

```
FastIron(config)#lldp med network-policy application voice tagged vlan 99 priority 3
dscp 22 port e 2/6
```

The network policy advertisement will appear similar to the following on the remote device, and in the CLI display output on the Foundry device (**show lldp local-info**):

```
+ MED Network Policy
 Application Type  : Voice
 Policy Flags      : Known Policy, Tagged
 VLAN ID           : 99
 L2 Priority       : 3
 DSCP Value        : 22
```

NOTE: Endpoints will advertise a policy as "unknown" in the **show lldp neighbor detail** command output, if it is a policy that is required by the Endpoint and the Endpoint has not yet received it.

Configuration Syntax

The CLI syntax for defining an LLDP-MED network policy differs for tagged, untagged, and priority tagged traffic. Refer to the appropriate syntax, below.

For tagged traffic:

Syntax: [no] lldp med network-policy application <application type> tagged vlan <vlan ID> priority <0 – 7> dscp <0 – 63> ports ethernet <port-list> | all

For untagged traffic:

Syntax: [no] lldp med network-policy application <application type> untagged dscp <0 – 63> ports ethernet <port-list> | all

For priority-tagged traffic:

Syntax: [no] lldp med network-policy application <application type> priority-tagged priority <0 – 7> dscp <0 – 63> ports ethernet <port-list> | all

For <port-list>, specify the port(s) in one of the following formats:

- FastIron GS and LS compact switches – <stacknum/slotnum/portnum>
- FastIron chassis devices – <slotnum/portnum>
- FESX, and FWSX compact switches – <portnum>

You can list all of the ports individually, use the keyword **to** to specify ranges of ports, or a combination of both. To apply the configuration to all ports on the device, use the keyword **all** instead of listing the ports individually.

<application type> indicates the primary function of the application(s) defined by this network policy. Application type can be one of the following:

- **guest-voice** – Limited voice service for guest users and visitors with their own IP telephony handsets or similar devices that support interactive voice services.
- **guest-voice-signaling** – Limited voice service for use in network topologies that require a different policy for guest voice signaling than for guest voice media.
- **softphone-voice** – Softphone voice service for use with multi-media applications that work in association with VoIP technology, enabling phone calls direct from a PC or laptop. Softphones do not usually support multiple VLANs, and are typically configured to use an untagged VLAN or a single tagged data-specific VLAN. Note that when a network policy is defined for use with an untagged VLAN, the Layer 2 priority field is ignored and only the DSCP value is relevant.
- **streaming-video** – Applies to broadcast- or multicast-based video content distribution and similar applications that support streaming video services requiring specific network policy treatment. Video applications that rely on TCP without buffering would not be an intended use of this application type.
- **video-conferencing** – Applies to dedicated video conferencing equipment and similar devices that support real-time interactive video/audio services.
- **video-signaling** – For use in network topologies that require a separate policy for video signaling than for video media. Note that this application type should not be advertised if all the same network policies apply as those advertised in the video conferencing policy TLV.
- **voice** – For use by dedicated IP telephony handsets and similar devices that support interactive voice services.
- **voice-signaling** – For use in network topologies that require a different policy for voice signaling than for voice media. Note that this application type should not be advertised if all the same network policies apply as those advertised in the voice policy TLV.

tagged vlan <vlan id> specifies the tagged VLAN that the specified application type will use.

untagged indicates that the device is using an untagged frame format.

priority-tagged indicates that the device uses priority-tagged frames. In this case, the device uses the default VLAN (PVID) of the ingress port.

priority <0 –7> indicates the Layer 2 priority value to be used for the specified application type. Enter 0 to use the default priority.

dscp <0 – 63> specifies the Layer 3 Differentiated Service codepoint priority value to be used for the specified application type. Enter 0 to use the default priority.

For <port-list>, specify the port(s) in one of the following formats:

- FastIron GS and LS compact switches – <stacknum/slotnum/portnum>
- FastIron chassis devices – <slotnum/portnum>
- FESX, and FWSX compact switches – <portnum>

You can list all of the ports individually, use the keyword **to** to specify ranges of ports, or a combination of both. To apply the configuration to all ports on the device, use the keyword **all** instead of listing the ports individually. Note that using the keyword **all** may cause undesirable effects on some ports. For example, if you configure **all** ports to advertise their VLAN name, and the configuration includes ports that are not members of any VLAN, the system will warn of the misconfigurations on non-member VLAN ports. The configuration will be applied to all ports, however, the ports that are not members of any VLAN will not send VLAN name advertisements.

LLDP-MED Attributes Advertised by the Foundry Device

LLDP-MED attributes are only advertised on a port if LLDP-MED is enabled (which is done by enabling the LLDP-MED capabilities TLV), the port's operating mode is *receive and transmit* (the default), and the port has received an LLDP-MED advertisement from an Endpoint. By default, the Foundry device will automatically advertise the following LLDP-MED attributes when the above criteria are met:

- LLDP-MED capabilities
- Location ID
- Network policy
- Power-via-MDI information

NOTE: Although the Location ID and Network policy attributes are automatically advertised, they will have no effect until they are actually defined.

LLDP-MED Capabilities

When enabled, LLDP-MED is enabled, and the LLDP-MED capabilities TLV is sent whenever any other LLDP-MED TLV is sent. When disabled, LLDP-MED is disabled and no LLDP-MED TLVs are sent.

The LLDP-MED capabilities advertisement includes the following information:

- The supported LLDP-MED TLVs
- The device type (Network Connectivity device or Endpoint (Class 1, 2, or 3))

By default, LLDP-MED information is automatically advertised when LLDP-MED is enabled. To disable this advertisement, enter a command such as the following:

```
FastIron(config)#no lldp advertise med-capabilities ports e 2/4 to 2/12
```

NOTE: Disabling the LLDP-MED capabilities TLV disables LLDP-MED.

To re-enable the LLDP-MED Capabilities TLV (and LLDP-MED) after it has been disabled, enter a command such as the following:

```
FastIron(config)#lldp advertise med-capabilities ports e 2/4 to 2/12
```

The LLDP-MED capabilities advertisement will appear similar to the following on the remote device, and in the CLI display output on the Foundry device (**show lldp local-info**):

```
+ MED capabilities: capabilities, networkPolicy, location, extendedPSE
  MED device type : Network Connectivity
```

Syntax: [no] lldp advertise med-capabilities ports ethernet <port-list> | all

For <port-list>, specify the port(s) in one of the following formats:

- FastIron GS and LS compact switches – <stacknum/slotnum/portnum>
- FastIron chassis devices – <slotnum/portnum>
- FESX, and FWSX compact switches – <portnum>

You can list all of the ports individually, use the keyword **to** to specify ranges of ports, or a combination of both. To apply the configuration to all ports on the device, use the keyword **all** instead of listing the ports individually. Note that using the keyword **all** may cause undesirable effects on some ports. For example, if you configure **all** ports to advertise their VLAN name, and the configuration includes ports that are not members of any VLAN, the system will warn of the misconfigurations on non-member VLAN ports. The configuration will be applied to all ports, however, the ports that are not members of any VLAN will not send VLAN name advertisements.

Extended Power-via-MDI Information

The extended Power-via-MDI TLV enables advanced power management between LLDP-MED Endpoints and Network Connectivity Devices. This TLV provides significantly more information than the 802.1AB Power-via-MDI TLV referenced in “Power-via-MDI” on page 25-22. For example, this TLV enables an Endpoint to communicate a more precise required power level, thereby enabling the device to allocate less power to the Endpoint, while making more power available to other ports.

The LLDP-MED Power-via-MDI TLV advertises an Endpoint's IEEE 802.3af power-related information, including the following:

- **Power type** – indicates whether the LLDP-MED device transmitting the LLPDU is a ***power sourcing device*** or a ***powered device***:
    - **Power sourcing device/equipment (PSE)** – This is the source of the power, or the device that integrates the power onto the network. Power sourcing devices/equipment have embedded POE technology. In this case, the power sourcing device is Foundry's POE device.
    - **Powered device (PD)** – This is the Ethernet device that requires power and is situated on the other end of the cable opposite the power sourcing device.
- **Power source** – The power source being utilized by a PSE or PD, for example, primary power source, backup power source, or unknown.

  For Endpoint devices, the power source information indicates the power capability of the Network Connectivity Device it is attached to. When the Network Connectivity device advertises that it is using its primary power source, the Endpoint should expect to have uninterrupted access to its available power. Likewise, if the Network Connectivity device advertises that it is using backup power, the Endpoint should not expect continuous power. The Endpoint may additionally choose to power down non-essential subsystems or to conserve power as long as the PSE is advertising that it is operating on backup power.

  NOTE: Foundry devices always advertise the power source as "unknown".

- **Power priority** – The in-line power priority level for the PSE or PD:
    - 3 – low
    - 2 – high
    - 1 – critical
    - unknown
- **Power level** – The total power, in tenths of watts, required by a PD from a PSE, or the total power a PSE is capable of sourcing over a maximum length cable based on its current configuration.

  If the exact power is not known for a PSE or PD, it will advertise the power level associated with its 802.3af power class (listed in Table 25.6).

Table 25.6: 802.3af Power Classes

Power Class	Minimum Power Level Output at the PSE	Maximum Power Levels at the PD
0	15.4 watts	0.44 – 12.95 watts
1	4.0 watts	0.44 – 3.84 watts
2	7.0 watts	3.84 – 6.49 watts
3	15.4 watts	6.49 – 12.95 watts

For a PD (Endpoint device), the power level represents the maximum power it can consume during normal operations in its current configuration, even if its actual power draw at that instance is less than the advertised power draw.

For a PSE (Network Connectivity device), the power level represents the amount of power that is available on the port at the time. If the PSE is operating in reduced power (i.e., it is using backup power), the reduced power capacity is advertised as long as the condition persists.

By default, LLDP-MED power-via-MDI information is automatically advertised when LLDP-MED is enabled, the port is a POE port, and POE is enabled on the port. To disable this advertisement, enter a command such as the following:

```
FastIron(config)#no lldp advertise med-power-via-mdi ports e 2/4 to 2/12
```

The LLDP-MED power-via-MDI advertisement will appear similar to the following on the remote device, and in the CLI display output on the Foundry device (**show lldp local-info**):

```
+ MED Extended Power via MDI
   Power Type     : PSE device
   Power Source   : Unknown Power Source
   Power Priority : Low (3)
   Power Value    : 6.5 watts (PSE equivalent: 7005 mWatts)
```

Syntax: [no] lldp advertise med-power-via-mdi ports ethernet <port-list> | all

For <port-list>, specify the port(s) in one of the following formats:

- FastIron GS and LS compact switches – <stacknum/slotnum/portnum>
- FastIron chassis devices – <slotnum/portnum>
- FESX, and FWSX compact switches – <portnum>

You can list all of the ports individually, use the keyword **to** to specify ranges of ports, or a combination of both. To apply the configuration to all ports on the device, use the keyword **all** instead of listing the ports individually. Note that using the keyword **all** may cause undesirable effects on some ports. For example, if you configure **all** ports to advertise their VLAN name, and the configuration includes ports that are not members of any VLAN, the system will warn of the misconfigurations on non-member VLAN ports. The configuration will be applied to all ports, however, the ports that are not members of any VLAN will not send VLAN name advertisements.

Displaying LLDP Statistics and Configuration Settings

You can use the following CLI **show** commands to display information about LLDP settings and statistics:

- **show lldp** – Displays a summary of the LLDP configuration settings.
- **show lldp statistics** – Displays LLDP global and per-port statistics.
- **show lldp neighbors** – Displays a list of the current LLDP neighbors.
- **show lldp neighbors detail** – Displays the details of the latest advertisements received from LLDP neighbors.
- **show lldp local-info** – Displays the details of the LLDP advertisements that will be transmitted on each port.

This above **show** commands are described in this section.

LLDP Configuration Summary

To display a summary of the LLDP configuration settings on the device, enter the **show lldp** command at any level of the CLI.

The following shows an example report.

```
FastIron#show lldp

LLDP transmit interval           : 10 seconds
LLDP transmit hold multiplier    : 4  (transmit TTL: 40 seconds)
LLDP transmit delay              : 1 seconds
LLDP SNMP notification interval  : 5 seconds
LLDP reinitialize delay          : 1 seconds
LLDP-MED fast start repeat count : 3

LLDP maximum neighbors           : 392
LLDP maximum neighbors per port  : 4
```

Syntax: show lldp

The following table describes the information displayed by the **show lldp statistics** command.

This Field...	Displays...
LLDP transmit interval	The number of seconds between regular LLDP packet transmissions.
LLDP transmit hold multiplier	The multiplier used to compute the actual time-to-live (TTL) value of an LLDP advertisement. The TTL value is the transmit interval multiplied by the transmit hold multiplier.
LLDP transmit delay	The number of seconds the LLDP agent will wait after transmitting an LLDP frame and before transmitting another LLDP frame.
LLDP SNMP notification interval	The number of seconds between transmission of SNMP LLDP traps (lldpRemTablesChange) and SNMP LLDP-MED traps (lldpXMedTopologyChangeDetected).
LLDP reinitialize delay	The minimum number of seconds the device will wait from when LLDP is disabled on a port, until a request to re-enable LLDP on that port will be honored.
LLDP-MED fast start repeat count	The number of seconds between LLDP frame transmissions when an LLDP-MED Endpoint is newly detected.
LLDP maximum neighbors	The maximum number of LLDP neighbors for which LLDP data will be retained, per device.
LLDP maximum neighbors per port	The maximum number of LLDP neighbors for which LLDP data will be retained, per port.

LLDP Statistics

The **show lldp statistics** command displays an overview of LLDP neighbor detection on the device, as well as packet counters and protocol statistics. The statistics are displayed on a global basis.

The following shows an example report.

```
FastIron#show lldp statistics

Last neighbor change time: 23 hours 50 minutes 40 seconds ago

Neighbor entries added          : 14
Neighbor entries deleted        : 5
Neighbor entries aged out       : 4
Neighbor advertisements dropped : 0

Port       Tx Pkts   Rx Pkts   Rx Pkts   Rx Pkts   Rx TLVs   Rx TLVs Neighbors
             Total     Total  w/Errors Discarded Unrecognz Discarded  Aged Out
1            60963     75179         0         0         0         0         4
2                0         0         0         0         0         0         0
3            60963     60963         0         0         0         0         0
4            60963    121925         0         0         0         0         0
5                0         0         0         0         0         0         0
6                0         0         0         0         0         0         0
7                0         0         0         0         0         0         0
8                0         0         0         0         0         0         0
9                0         0         0         0         0         0         0
10           60974         0         0         0         0         0         0
11               0         0         0         0         0         0         0
12               0         0         0         0         0         0         0
13               0         0         0         0         0         0         0
14               0         0         0         0         0         0         0
```

Syntax: show lldp statistics

NOTE: You can reset LLDP statistics using the CLI command **clear LLDP statistics**. See “Resetting LLDP Statistics” on page 25-44.

The following table describes the information displayed by the **show lldp statistics** command.

This Field...	Displays...
Last neighbor change time	The elapsed time (in hours, minutes, and seconds) since a neighbor last advertised information. For example, the elapsed time since a neighbor was last added, deleted, or its advertised information changed.
Neighbor entries added	The number of new LLDP neighbors detected since the last reboot or since the last time the **clear lldp statistics all** command was issued.
Neighbor entries deleted	The number of LLDP neighbors deleted since the last reboot or since the last time the **clear lldp statistics all** command was issued.

This Field...	Displays...
Neighbor entries aged out	The number of LLDP neighbors dropped on all ports after the time-to-live expired. Note that LLDP entries age out naturally when a port's cable or module is disconnected or when a port becomes disabled. However, if a disabled port is re-enabled, the system will delete the old LLDP entries.
Neighbor advertisements dropped	The number of valid LLDP neighbors the device detected, but could not add. This can occur, for example, when a new neighbor is detected and the device is already supporting the maximum number of neighbors possible. This can also occur when an LLDPDU is missing a mandatory TLV or is not formatted correctly.
Port	The local port number.
Tx Pkts Total	The number of LLDP packets the port transmitted.
Rx Pkts Total	The number of LLDP packets the port received.
Rx Pkts w/Errors	The number of LLDP packets the port received that have one or more detectable errors.
Rx Pkts Discarded	The number of LLDP packets the port received then discarded.
Rx TLVs Unrecognz	The number of TLVs the port received that were not recognized by the LLDP local agent. Unrecognized TLVs are retained by the system and can be viewed in the output of the **show LLDP neighbors detail** command or retrieved via SNMP.
Rx TLVs Discarded	The number of TLVs the port received then discarded.
Neighbors Aged Out	The number of times a neighbor's information was deleted because its TTL timer expired.

LLDP Neighbors

The **show lldp neighbors** command displays a list of the current LLDP neighbors per port.

The following shows an example report.

```
FastIron#show lldp neighbors

Lcl Port Chassis ID      Port ID         Port Description        System Name
1        0004.1234.0fc0  0004.1234.0fc0  GigabitEthernet9/1      FastIron Supe~
1        00e0.5201.4000  00e0.5201.4000  GigabitEthernet0/1/1    FGS624XGP Swi~
3        00e0.5211.0200  00e0.5211.0203  GigabitEthernet4        FESX424+2XG S~
4        00e0.5211.0200  00e0.5211.0202  GigabitEthernet3        FESX424+2XG S~
4        00e0.5211.0200  00e0.5211.0210  GigabitEthernet17       FESX424+2XG S~
15       00e0.5211.0200  00e0.5211.020f  GigabitEthernet16       FESX424+2XG S~
16       00e0.5211.0200  00e0.5211.020e  GigabitEthernet15       FESX424+2XG S~
17       00e0.5211.0200  00e0.5211.0211  GigabitEthernet18       FESX424+2XG S~
18       00e0.5211.0200  00e0.5211.0210  GigabitEthernet17       FESX424+2XG S~
```

Syntax: show lldp neighbors

The following table describes the information displayed by the **show lldp neighbors** command.

This Field...	Displays...
Lcl Port	The local LLDP port number.

This Field...	Displays...
Chassis ID	The identifier for the chassis. Foundry devices use the base MAC address of the device as the Chassis ID.
Port ID	The identifier for the port. Foundry devices use the permanent MAC address associated with the port as the port ID.
Port Description	The description for the port. Foundry devices use the ifDescr MIB object from MIB-II as the port description.
System Name	The administratively-assigned name for the system. Foundry devices use the sysName MIB object from MIB-II, which corresponds to the CLI **hostname** command setting. **NOTE:** A tilde (~) at the end of a line indicates that the value in the field is too long to display in full and is truncated.

LLDP Neighbors Detail

The **show lldp neighbors detail** command displays the LLDP advertisements *received* from LLDP neighbors.

The following shows an example **show lldp neighbors detail** report.

NOTE: The **show lldp neighbors detail** output will vary depending on the data received. Also, values that are not recognized or do not have a recognizable format, may be displayed in hexadecimal binary form.

```
FastIron#show lldp neighbors detail ports e 1/9

Local port: 1/9
  Neighbor: 0800.0f18.cc03, TTL 101 seconds
    + Chassis ID (network address): 10.43.39.151
    + Port ID (MAC address): 0800.0f18.cc03
    + Time to live: 120 seconds
    + Port description    : "LAN port"
    + System name         : "regDN 1015,MITEL 5235 DM"
    + System description  : "regDN 1015,MITEL 5235 DM,h/w rev 2,ASIC rev 1,f/w\
                             Boot 02.01.00.11,f/w Main 02.01.00.11"
    + System capabilities : bridge, telephone
      Enabled capabilities: bridge, telephone
    + Management address (IPv4): 10.43.39.151
    + 802.3 MAC/PHY          : auto-negotiation enabled
      Advertised capabilities: 10BaseT-HD, 10BaseT-FD, 100BaseTX-HD,
                               100BaseTX-FD
      Operational MAU type   : 100BaseTX-FD
    + MED capabilities: capabilities, networkPolicy, extendedPD
      MED device type : Endpoint Class III
    + MED Network Policy
      Application Type  : Voice
      Policy Flags      : Known Policy, Tagged
      VLAN ID           : 300
      L2 Priority       : 7
      DSCP Value        : 7
    + MED Extended Power via MDI
      Power Type     : PD device
      Power Source   : Unknown Power Source
      Power Priority : High (2)
      Power Value    : 6.2 watts (PSE equivalent: 6656 mWatts)
    + MED Hardware revision : "PCB Version: 2"
    + MED Firmware revision : "Boot 02.01.00.11"
    + MED Software revision : "Main 02.01.00.11"
    + MED Serial number     : ""
    + MED Manufacturer      : "Mitel Corporation"
    + MED Model name        : "MITEL 5235 DM"
    + MED Asset ID          : ""
```

A backslash (\) at the end of a line indicates that the text continues on the next line.

Except for the following field, the fields in the above output are described in the individual TLV advertisement sections in this chapter.

This Field...	Displays...
Neighbor	The source MAC address from which the packet was received, and the remaining TTL for the neighbor entry.

Syntax: show lldp neighbors detail [ports ethernet <port-list> | all]

If you do not specify any ports or use the keyword **all**, by default, the report will show the LLDP neighbor details for all ports.

For <port-list>, specify the port(s) in one of the following formats:

- FastIron GS and LS compact switches – <stacknum/slotnum/portnum>
- FastIron chassis devices – <slotnum/portnum>
- FESX, and FWSX compact switches – <portnum>

You can list all of the ports individually, use the keyword **to** to specify ranges of ports, or a combination of both. To apply the configuration to all ports on the device, use the keyword **all** instead of listing the ports individually.

LLDP Configuration Details

The **show lldp local-info** command displays the local information advertisements (TLVs) that will be *transmitted* by the LLDP agent.

NOTE: The **show lldp local-info** output will vary based on LLDP configuration settings.

The following shows an example report.

```
FastIron#show lldp local-info ports e 20

Local port: 20
  + Chassis ID (MAC address): 0012.f233.e2c0
  + Port ID (MAC address): 0012.f233.e2d3
  + Time to live: 40 seconds
  + System name: "FESX424_POE"
  + Port description: "GigabitEthernet20"
  + System description  : "Foundry Networks, Inc. FESX424-PREM-PoE, IronWare V\
                          ersion 04.0.00b256T3e1 Compiled on Sep 04 2007 at 0\
                          3:54:29 labeled as SXS04000b256"
  + System capabilities : bridge
    Enabled capabilities: bridge
  + 802.3 MAC/PHY       : auto-negotiation enabled
    Advertised capabilities: 10BaseT-HD, 10BaseT-FD, 100BaseTX-HD,
                             100BaseTX-FD, fdxSPause, fdxBPause, 1000BaseT-HD,
                             1000BaseT-FD
    Operational MAU type: 100BaseTX-FD
  + 802.3 Power via MDI: PSE port, power enabled, class 2
    Power Pair         : A (not controllable)
  + Link aggregation: not capable
  + Maximum frame size: 1522 octets
  + MED capabilities: capabilities, networkPolicy, location, extendedPSE
    MED device type : Network Connectivity
  + MED Network Policy
    Application Type  : Voice
    Policy Flags      : Known Policy, Tagged
    VLAN ID           : 99
    L2 Priority       : 3
    DSCP Value        : 22
  + MED Network Policy
    Application Type  : Video Conferencing
    Policy Flags      : Known Policy, Tagged
    VLAN ID           : 100
    L2 Priority       : 5
    DSCP Value        : 10
  + MED Location ID
    Data Format: Coordinate-based location
    Latitude Resolution  : 20 bits
    Latitude Value       : -78.303 degrees
    Longitude Resolution : 18 bits
    Longitude Value      : 34.27 degrees
    Altitude Resolution  : 16 bits
    Altitude Value       : 50. meters
    Datum                : WGS 84
```

(cont'd from previous page)....

```
+ MED Location ID
    Data Format: Civic Address
    Location of: Client
    Country     : "US"
    CA Type     : 1
    CA Value    : "CA"
    CA Type     : 3
    CA Value    : "Santa Clara"
    CA Type     : 6
    CA Value    : "4980 Great America Pkwy."
    CA Type     : 24
    CA Value    : "95054"
    CA Type     : 27
    CA Value    : "5"
    CA Type     : 28
    CA Value    : "551"
    CA Type     : 29
    CA Value    : "office"
    CA Type     : 23
    CA Value    : "John Doe"
  + MED Location ID
    Data Format: ECS ELIN
    Value       : "1234567890"
  + MED Extended Power via MDI
    Power Type     : PSE device
    Power Source   : Unknown Power Source
    Power Priority : Low (3)
    Power Value    : 6.5 watts (PSE equivalent: 7005 mWatts) + Port VLAN ID: 99
  + Management address (IPv4): 192.1.1.121
  + VLAN name (VLAN 99): "Voice-VLAN-99"
```

A backslash (\) at the end of a line indicates that the text continues on the next line.

The fields in the above output are described in the individual TLV advertisement sections in this chapter.

Syntax: show lldp local-info [ports ethernet <port-list> | all]

If you do not specify any ports or use the keyword **all**, by default, the report will show the local information advertisements for all ports.

For <port-list>, specify the port(s) in one of the following formats:

- FastIron GS and LS compact switches – <stacknum/slotnum/portnum>
- FastIron chassis devices – <slotnum/portnum>
- FESX, and FWSX compact switches – <portnum>

You can list all of the ports individually, use the keyword **to** to specify ranges of ports, or a combination of both. To apply the configuration to all ports on the device, use the keyword **all** instead of listing the ports individually.

Resetting LLDP Statistics

To reset LLDP statistics, enter the **clear lldp statistics** command at the Global CONFIG level of the CLI. The Foundry device will clear the global and per-port LLDP neighbor statistics on the device (see "LLDP Statistics" on page 25-38).

```
FastIron#clear lldp statistics
```

Syntax: clear lldp statistics [ports ethernet <port-list> | all]

If you do not specify any ports or use the keyword **all**, by default, the system will clear lldp statistics on all ports.

For <port-list>, specify the port(s) in one of the following formats:

- FastIron GS and LS compact switches – <stacknum/slotnum/portnum>
- FastIron chassis devices – <slotnum/portnum>
- FESX, and FWSX compact switches – <portnum>

You can list all of the ports individually, use the keyword **to** to specify ranges of ports, or a combination of both. To apply the configuration to all ports on the device, use the keyword **all** instead of listing the ports individually.

Clearing Cached LLDP Neighbor Information

The Foundry device clears cached LLDP neighbor information after a port becomes disabled and the LLDP neighbor information ages out. However, if a port is disabled then re-enabled before the neighbor information ages out, the device will clear the cached LLDP neighbor information when the port is re-enabled.

If desired, you can manually clear the cache. For example, to clear the cached LLDP neighbor information for port e 20, enter the following command at the Global CONFIG level of the CLI.

```
FastIron#clear lldp neighbors ports e 20
```

Syntax: clear lldp neighbors [ports ethernet <port-list> | all]

If you do not specify any ports or use the keyword **all**, by default, the system will clear the cached LLDP neighbor information for all ports.

For <port-list>, specify the port(s) in one of the following formats:

- FastIron GS and LS compact switches – <stacknum/slotnum/portnum>
- FastIron chassis devices – <slotnum/portnum>
- FESX, and FWSX compact switches – <portnum>

You can list all of the ports individually, use the keyword **to** to specify ranges of ports, or a combination of both. To apply the configuration to all ports on the device, use the keyword **all** instead of listing the ports individually.

Chapter 26

Configuring IP Multicast Traffic Reduction for the FastIron GS and FastIron LS

This chapter describes how to configure IP Multicast Traffic Reduction (IGMP Snooping), and PIM SM Traffic Snooping parameters on Foundry FastIron GS and FastIron LS devices.

NOTE: This chapter applies to FastIron GS and FastIron LS devices only. For information about configuring IP Multicast Traffic Reduction on other FastIron devices, see “Configuring IP Multicast Traffic Reduction for the FastIron X Series Switch” on page 27-1.

IGMP Snooping for FastIron GS and FastIron LS Devices

Platform Support:

- FGS and FLS devices running software release 03.0.00 and later

IGMP Snooping Overview

When a device processes a multicast packet, by default, the device broadcasts the packets to all ports except the incoming port of a VLAN. Packets are flooded by hardware without going to the CPU. This behavior causes some clients to receive unwanted traffic.

IGMP snooping provides multicast containment by forwarding traffic to only the ports that have IGMP receivers for a specific multicast group (destination address). A device maintains the IGMP group membership information by processing the IGMP reports and leave messages, so traffic can be forwarded to ports receiving IGMP reports.

An IPv4 multicast address is a destination address in the range of 224.0.0.0 to 239.255.255.255. Addresses of 224.0.0.X are reserved. Because packets destined for these addresses may require VLAN flooding, devices do not do snooping in the reserved range. Data packets destined to addresses in reserved range are flooded to the entire VLAN by hardware, and mirrored to the CPU. Multicast data packets destined for the non-reserved range of addresses are snooped. A client must send IGMP reports in order to receive traffic. If an application outside the reserved range requires VLAN flooding, the user must configure a static group that applies to the entire VLAN. In addition, a static group with the drop option can discard multicast data packets to a specified group in hardware, including addresses in the reserved range.

An IGMP device's responsibility is to broadcast general queries periodically, and to send group queries when receiving a leave message, to confirm that none of the clients on the port still want specific traffic before removing the traffic from the port. IGMPv2 lets clients specify what group (destination address) will receive the traffic but not to specify the source of the traffic. IGMPv3 is for source-specific multicast traffic, adding the capability for clients to INCLUDE or EXCLUDE specific traffic sources. An IGMPv3 device port state could be INCLUDE or EXCLUDE, and there are different types of group records for client reports.

The interfaces respond to general or group queries by sending a membership report that contains one or more of the following records associated with a specific group:

- Current-state record that indicates from which sources the interface wants to receive and not receive traffic. This record contains the source address of interfaces and whether or not traffic will be included (IS_IN) or not excluded (IS_EX) from this source.
- Filter-mode-change record. If the interface state changes from IS_IN to IS_EX, a TO_EX record is included in the membership report. Likewise, if the interface state changes from IS_EX to IS_IN, a TO_IN record appears in the membership report.
- An IGMPv2 leave report is equivalent to a TO_IN (empty) record in IGMPv3. This record means that no traffic from this group will be received regardless of the source.
- An IGMPv2 group report is equivalent to an IS_EX (empty) record in IGMPv3. This record means that all traffic from this group will be received regardless of source.
- Source-list-change record. If the interface wants to add or remove traffic sources from its membership report, the report can contain an ALLOW record, which includes a list of new sources from which the interface wishes to receive traffic. It can also contain a BLOCK record, which lists the current traffic sources from which the interface wants to stop receiving traffic.

IGMP protocols provide a method for clients and a device to exchange messages, and let the device build a database indicating which port wants what traffic. The protocols do not specify forwarding methods. They require IGMP snooping or multicast protocols such as PIM or DVMRP to handle packet forwarding. PIM and DVMRP can route multicast packets within and outside a VLAN, while IGMP snooping can switch packets only within a VLAN. Currently, FGS and FLS devices do not support multicast routing.

If a VLAN is not IGMP snooping-enabled, it floods multicast data and control packets to the entire VLAN in hardware. When snooping is enabled, IGMP packets are trapped to the CPU. Data packets are mirrored to the CPU in addition to being VLAN flooded. The CPU then installs hardware resources, so that subsequent data packets can be switched to desired ports in hardware without going to the CPU. If there is no client report or port to queriers for a data stream, the hardware resource drops it. The hardware can either match the group address only (* G), or both the source and group (S G) of the data stream. If any IGMPv3 is configured in any port of a VLAN, this VLAN uses (S G) match; otherwise, it uses (* G). This is 32-bit IP address matching, not 23-bit multicast MAC address 01-00-5e-xx-xx-xx matching.

An FGS or FLS device has 16K of hardware resources for MAC learning, IGMP, and MLD snooping. If a data packet does not match any of these resources, it might be sent to the CPU, which increases the CPU burden. This can happen if the device runs out of hardware resource, or is unable to install resources for a specific matching address due to hashing collision. The hardware hashes addresses into 16K entries, with some addresses hashed into the same entry. If the collision number in an entry is more than the hardware chain length, the resource cannot be installed. The chain length can be configured using the **hash-chain-length** command:

```
FastIron(config)#hash-chain-length 8
```

Syntax: [no] hash-chain-length <num>

The <num> parameter range is 4 to 32, in multiples of 4. If the input value is not a multiple of 4, then it will be changed to the multiple of 4 lower than then the input value (e.g. 11 will be changed to 8). The default hash chain length is 4. A chain length of more than 4 may affect line rate switching.

NOTE: For this command to take effect, you must save the configuration and reload the switch.

The hardware resource limit applies only to the VLANs where IGMP snooping is enabled. Multicast streams are switched in hardware without using any pre-installed resources in a VLAN where snooping is not enabled.

An FGS or FLS device supports up to 32K of IGMP groups, which are produced by client membership reports.

Configuration Notes

- Servers (traffic sources) are not required to send IGMP memberships.
- The default IGMP version is V2.

- Hardware resource is installed only when there is data traffic. If a VLAN is configured for IGMPv3, the hardware matches (S G), otherwise it matches (* G).
- A user can configure the maximum numbers of groups and hardware switched data streams.
- The device supports static groups that apply to the entire VLAN, or to just a few ports. The device acts as a proxy to send IGMP reports for the static groups when receiving queries. The static group has a drop option to discard multicast data packets in hardware.
- A user can configure static router ports to force all multicast traffic to these specific ports.
- The devices support fast leave for IGMPv2. Fast leave stops traffic immediately when the port receives a leave message.
- The devices support tracking and fast leave for IGMPv3, tracking all IGMPv3 clients. If the only client on a port leaves, traffic is stopped immediately.
- An IGMP device can be configured as a querier (active) or non-querier (passive). Queriers send queries. Non-queriers listen for queries and forward them to the entire VLAN.
- Every VLAN can be independently configured to be a querier or a non-querier.
- If a VLAN has a connection to a PIM/DVMRP-enabled port on another router, this VLAN should be configured as a non-querier (passive). When multiple snooping devices connect together and there is no connection to PIM/DVMRP ports, one device should be configured as a querier (active). If multiple devices are configured as active (queriers), only one will keep sending queries after exchanging queries.
- An IGMP device can be configured to rate-limit the forwarding IGMPv2 membership reports to queriers.
- The querier must configure an IP address to send out queries.

The implementation allows snooping on some VLANs or all VLANs. Each VLAN can independently enable or disable IGMP, or configure V2 or V3. In general, global configuration commands **ip multicast** apply to every VLAN except those that have local **multicast** configurations (which supersede the global configuration). IGMP also allows independent configuration of individual ports in a VLAN for either IGMPv2 or IGMPv3. Configuring a specific version on a port or a VLAN only applies to the device's sent queries. The device always processes client reports of any version regardless of the configured version.

IGMP snooping requires hardware resources. If resources are inadequate, the data stream without a resource is mirrored to CPU in addition to being VLAN flooded, which can cause high CPU usage. Foundry recommends that you avoid global enabling of snooping unless necessary.

When any port in a VLAN is configured for IGMPv3, the VLAN matches both source and group (S G) in hardware switching. If no ports in the VLAN are configured for IGMPv3, the VLAN matches group only (* G). Matching (S G) requires more hardware resources than matching (* G) when there are multiple servers sharing the same group. For example, two data streams from different sources to the same group require two (S G) entries in IGMPv3, but only one (* G) in IGMPv2. To conserve resources, IGMPv3 should be used only in source-specific applications. When VLANs are be independently configured for versions, some VLANs can match (* G) while others match (S G).

IGMP snooping requires clients to send membership reports in order to receive data traffic. If a client application does not send reports, you must configure static groups to force traffic to client ports. A static group can apply to only some ports or to the entire VLAN.

Configuring Queriers and Non-Queriers

An IGMP snooping-enabled FGS or FLS device can be configured as a querier (active) or non-querier (passive). An IGMP querier sends queries; a non-querier listens for IGMP queries and forwards them to the entire VLAN. Starting in FGS software release 03.0.00, VLANs can be independently configured to be queriers or non-queriers. If a VLAN has a connection to a PIM/DVMRP-enabled port on another router, the VLAN should be configured as a non-querier. When multiple IGMP snooping devices are connected together, and there is no connection to a PIM/DVMRP-enabled port, one of the devices should be configured as a querier. If multiple devices are configured as queriers, after these devices exchange queries, then all except the winner stop sending queries. The device with the lowest address becomes the querier. Although the system will work when multiple devices are configured as queriers, Foundry recommends that only one device (preferably the one with the traffic source) is configured as a querier.

The non-queriers always forward multicast data traffic and IGMP messages to router ports which receive IGMP queries or PIM/DVMRP hellos. Foundry recommends that you configure the device with the data traffic source (server) as a querier. If a server is attached to a non-querier, the non-querier always forwards traffic to the querier regardless of whether there are any clients on the querier.

NOTE: In a topology of one or more connecting devices, at least one device must be running PIM/DVMRP, or configured as active. Otherwise, none of the devices can send out queries, and traffic cannot be forwarded to clients.

VLAN Specific Configuration

You can configure IGMP snooping on some VLANs or on all VLANs. Each VLAN can be independently enabled or disabled for IGMP snooping, and can be configured for IGMPv2 or IGMPv3. In general, the **ip multicast** commands apply globally to all VLANs except those configured with VLAN-specific **multicast** commands. The VLAN-specific multicast commands supersede the global **ip multicast** commands.

Using IGMPv2 with IGMPv3

IGMP snooping can be configured for IGMPv2 or IGMPv3 on individual ports on a VLAN. An interface or router sends the queries and reports that include its IGMP version specified on it. The version configuration only applies to sending queries. The snooping device recognizes and processes IGMPv2 and IGMPv3 packets regardless of the version configuration.

To avoid version deadlock, an interface retains its version configuration even when it receives a report with a lower version.

PIM SM Traffic Snooping Overview

Platform Support:

- FGS and FLS devices running software release 02.6.00 and later

When multiple PIM sparse routers connect through a snooping-enabled device, the FGS or FLS device always forwards multicast traffic to these routers. For example, PIM sparse routers R1, R2 and R3 connect through a device. Assume R2 needs traffic, and R1 sends it to the device, which forwards it to both R2 and R3, even though R3 does not need it. A PIM snooping-enabled device listens to join and prune messages exchanged by PIM sparse routers, and stops traffic to the router that sends prune messages. This allows the device to forward the data stream to R2 only.

PIM SM traffic snooping requires IP multicast traffic reduction to be enabled on the device. IP multicast traffic reduction configures the device to listen for IGMP messages. PIM SM traffic snooping provides a finer level of multicast traffic control by configuring the device to listen specifically for PIM SM join and prune messages sent from one PIM SM router to another through the device.

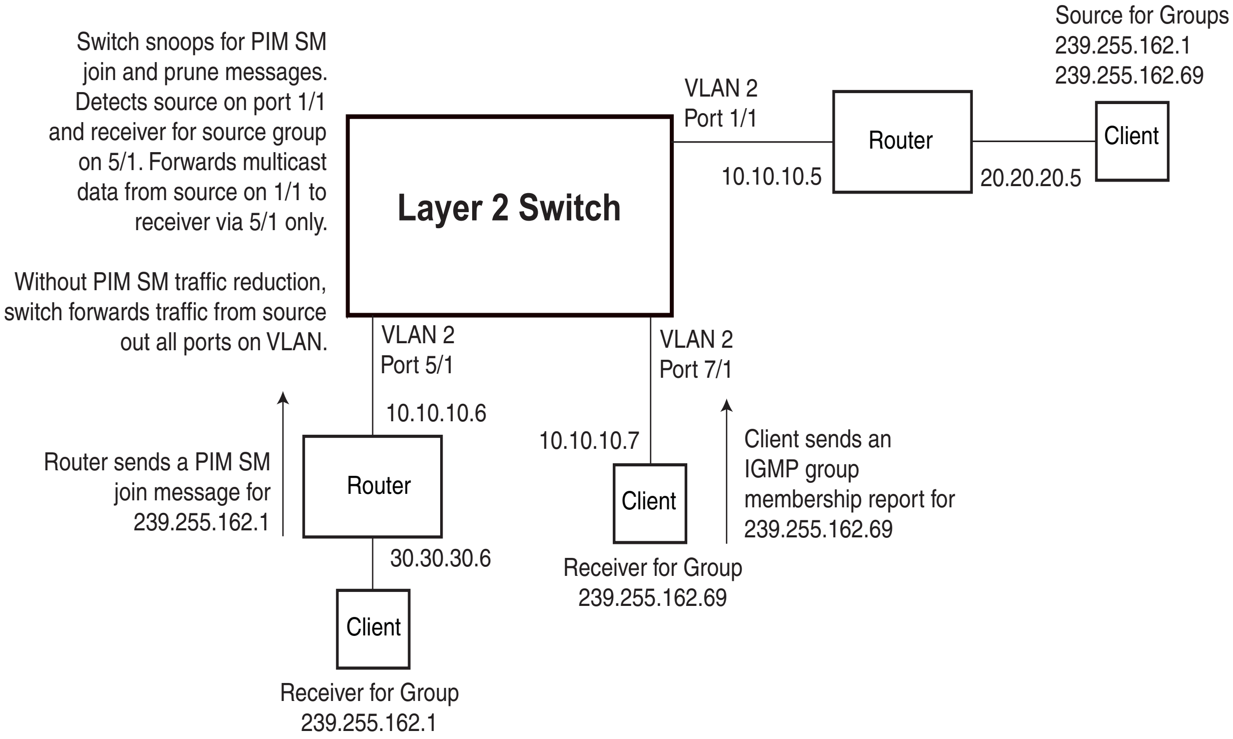
Configuration Notes

This feature applies only to PIM SM version 2 (PIM V2).

Application Example

Figure 26.1 shows an example application of the PIM SM traffic snooping feature. In this example, a device is connected through an IP router to a PIM SM group source that is sending traffic for two PIM SM groups. The device also is connected to a receiver for each of the groups.

Figure 26.1 PIM SM Traffic Reduction in an Enterprise Network


Switch snoops for PIM SM join and prune messages. Detects source on port 1/1 and receiver for source group on 5/1. Forwards multicast data from source on 1/1 to receiver via 5/1 only.
Without PIM SM traffic reduction, switch forwards traffic from source out all ports on VLAN.
Layer 2 Switch
VLAN 2
Port 1/1
10.10.10.5
Router
20.20.20.5
Client
Source for Groups
239.255.162.1
239.255.162.69
VLAN 2
Port 5/1
10.10.10.6
Router sends a PIM SM join message for 239.255.162.1
Router
30.30.30.6
Client
Receiver for Group
239.255.162.1
VLAN 2
Port 7/1
10.10.10.7
Client
Receiver for Group
239.255.162.69
Client sends an IGMP group membership report for 239.255.162.69

When PIM SM traffic snooping is enabled, the device starts listening for PIM SM join and prune messages and IGMP group membership reports. Until the device receives a PIM SM join message or an IGMP group membership report, the device forwards IP multicast traffic out all ports. Once the device receives a join message or group membership report for a group, the device forwards subsequent traffic for that group only on the ports from which the join messages or IGMP reports were received.

In this example, the router connected to the receiver for group 239.255.162.1 sends a join message toward the group's source. Since PIM SM traffic snooping is enabled on the device, the device examines the join message to learn the group ID, then makes a forwarding entry for the group ID and the port connected to the receiver's router. The next time the device receives traffic for 239.255.162.1 from the group's source, the device forwards the traffic only on port 5/1, since that is the only port connected to a receiver for the group.

Notice that the receiver for group 239.255.162.69 is directly connected to the device. As result, the device does not see a join message on behalf of the client. However, since IP multicast traffic reduction also is enabled, the device uses the IGMP group membership report from the client to select the port for forwarding traffic to group 239.255.162.69 receivers.

The IP multicast traffic reduction feature and the PIM SM traffic snooping feature together build a list of groups and forwarding ports for the VLAN. The list includes PIM SM groups learned through join messages as well as MAC addresses learned through IGMP group membership reports. In this case, even though the device never sees a join message for the receiver for group 239.255.162.69, the device nonetheless learns about the receiver and forwards group traffic to the receiver.

The device stops forwarding IP multicast traffic on a port for a group if the port receives a prune message for the group.

Notice that the ports connected to the source and the receivers are all in the same port-based VLAN on the device. This is required for the PIM SM snooping feature. The devices on the edge of the Global Ethernet cloud are configured for IP multicast traffic reduction and PIM SM traffic snooping. Although this application uses multiple devices, the feature has the same requirements and works the same way as it does on a single device.

Configuring IGMP Snooping

Configuring IGMP snooping on an FGS or FLS device consists of the following global and VLAN-specific tasks:

Global Tasks

- “Configuring the Hardware and Software Resource Limits” on page 26-6
- “Enabling or Disabling Transmission and Receipt of IGMP Packets on a Port” on page 26-7
- “Configuring the Global IGMP Mode” on page 26-7 (Must be enabled for IGMP snooping)
- “Modifying the Age Interval” on page 26-7
- “Modifying the Query Interval (Active IGMP Snooping Mode Only)” on page 26-7
- “Configuring the Global IGMP Version” on page 26-8
- “Configuring Report Control” on page 26-8 (rate limiting)
- “Modifying the Wait Time Before Stopping Traffic When Receiving a Leave Message” on page 26-8
- “Modifying the Multicast Cache Age Time” on page 26-8
- “Enabling or Disabling Error and Warning Messages” on page 26-8
- “Enabling or Disabling PIM Sparse Snooping” on page 26-9

VLAN-Specific Tasks

- “Configuring the IGMP Mode for a VLAN” on page 26-9 (active or passive)
- “Disabling IGMP Snooping for the VLAN” on page 26-9
- “Disabling PIM Sparse Mode Snooping for the VLAN” on page 26-9
- “Configuring the IGMP Version for the VLAN” on page 26-10
- “Configuring the IGMP Version for Individual Ports in the VLAN” on page 26-10
- “Configuring Static Groups to the Entire VLAN or to Specific Ports” on page 26-10
- “Configuring Static Router Ports” on page 26-10
- “Turning Off Static Group Proxy” on page 26-11
- “Enabling IGMPv3 Membership Tracking and Fast Leave for the VLAN” on page 26-11
- “Configuring Fast Leave for IGMPv2” on page 26-11
- “Enabling Fast Convergence” on page 26-11

Configuring the Hardware and Software Resource Limits

The system supports up to 8K of hardware-switched multicast streams. The configurable range is from 256 to 8192 with a default of 4096. Enter a command such as the following to define the maximum number of IGMP snooping cache entries:

```
FastIron(config)#system-max igmp-snoop-mcache 8000
```

Syntax: [no] system-max igmp-snoop-mcache <num>

The system supports up to 32K of groups. The configurable range is 256 to 32768 and the default is 8192. The configured number is the upper limit of an expandable database. Client memberships exceeding the group limits are not processed. Enter a command such as the following to define the maximum number of IGMP group addresses:

```
FastIron(config)#system-max igmp-max-group-addr 1600
```

Syntax: [no] system-max igmp-max-group-addr <num>

Enabling or Disabling Transmission and Receipt of IGMP Packets on a Port

When a VLAN is snooping-enabled, all IGMP packets are trapped to CPU without hardware VLAN flooding. The CPU can block IGMP packets to and from a multicast-disabled port, and does not add it to the output interfaces of hardware resources. This prevents the disabled port from receiving multicast traffic. However, if static groups to the entire VLAN are defined, the traffic from these groups is VLAN flooded, including to disabled ports. Traffic from disabled ports cannot be blocked in hardware, and is switched in the same way as traffic from enabled ports.

This command has no effect on a VLAN that is not snooping-enabled because all multicast traffic is VLAN flooded.

```
FastIron(config)#interface ethernet 0/1/3
```

```
FastIron(config-if-e1000-0/1/3)#ip-multicast-disable
```

Syntax: [no] ip-multicast-disable

Configuring the Global IGMP Mode

You can configure active or passive IGMP modes on the FGS or FLS device. The default mode is passive. If you specify an IGMP mode for a VLAN, it overrides the global setting.

- Active - When active IGMP mode is enabled, an FGS or FLS device actively sends out IGMP queries to identify multicast groups on the network, and makes entries in the IGMP table based on the group membership reports it receives.
- Passive - When passive IGMP mode is enabled, it forwards reports to the router ports which receive queries. IGMP snooping in the passive mode does not send queries. However, it forwards queries to the entire VLAN.

To globally set the IGMP mode to active, enter the following command:

```
FastIron(config)#ip multicast active
```

Syntax: [no] ip multicast [active | passive]

If you do not enter either *active* or *passive*, the passive mode is assumed.

Modifying the Age Interval

When the device receives a group membership report, it makes an entry for that group in the IGMP group table. The age interval specifies how long the entry can remain in the table before the device receives another group membership report. When multiple devices connect together, all devices should be configured for the same age interval, which should be at least twice the length of the query interval, so that missing one report won't stop traffic. Non-querier age intervals should be the same as the age interval of the querier.

To modify the age interval, enter a command such as the following:

```
FastIron(config)#ip multicast age-interval 280
```

Syntax: [no] ip multicast age-interval <interval>

The <interval> parameter specifies the aging time. You can specify a value from 20 - 7200 seconds. The default is 140 seconds.

Modifying the Query Interval (Active IGMP Snooping Mode Only)

For a device with an active IGMP mode, you can modify the query interval to specify how often the device sends group membership queries. When multiple queriers connect together, they should all be configured with the same query interval.

To modify the query interval, enter a command such as the following:

```
FastIron(config)#ip multicast query-interval 120
```

Syntax: [no] ip multicast query-interval <interval>

The <interval> parameter specifies the time between queries. You can specify a value from 10 - 3600 seconds. The default is 60 seconds.

Configuring the Global IGMP Version

You can globally specify IGMPv2 or IGMPv3 for the device. The default is IGMPv2. For example, the following command causes the device to use IGMPv3:

```
FastIron(config)#ip multicast version 3
```

Syntax: [no] ip multicast version 2|3

You can also optionally specify the IGMP version for individual VLANs, or individual ports within VLANs. When no IGMP version is specified for a VLAN, the global IGMP version is used. When an IGMP version is specified for individual ports within a VLAN, the ports use that version, instead of the VLAN version or the global version. The default is IGMPv2.

Configuring Report Control

A device in passive mode forwards reports and leave messages from clients to the upstream router ports that are receiving queries.

You can configure report control to rate-limit report forwarding within the same group to no more than once every 10 seconds. This rate-limiting does not apply to the first report answering a group-specific query.

NOTE: This feature applies to IGMPv2 only. The leave messages are not rate limited.

IGMPv2 membership reports of the same group from different clients are considered to be the same and are rate-limited.

Use the following command to alleviate report storms from many clients answering the upstream router query.

```
 FastIron(config)#ip multicast report-control
```

Syntax: [no] ip multicast report-control

The original command, **ip igmp-report-control,** has been renamed to **ip multicast report-control**. The original command is still accepted; however, it is renamed when you issue a **show configuration** command.

Modifying the Wait Time Before Stopping Traffic When Receiving a Leave Message

You can define the wait time before stopping traffic to a port when a leave message is received. The device sends group-specific queries once per second to ask if any client in the same port still needs this group. The value range is from 1 to 5, and the default is 2. Due to internal timer granularity, the actual wait time is between n and (n+1) seconds (n is the configured value).

```
FastIron(config)#ip multicast leave-wait-time 1
```

Syntax: [no] ip multicast leave-wait-time <num>

Modifying the Multicast Cache Age Time

You can set the time for an mcache to age out when it does not receive traffic. The traffic is hardware switched. One minute before aging out an mcache, the device mirrors a packet of this mcache to CPU to reset the age. If no data traffic arrives within one minute, this mcache is deleted. A lower value quickly removes resources consumed by idle streams, but it mirrors packets to CPU often. A higher value is recommended only data streams are continually arriving. The range is 60 to 3600 seconds, and the default is 60 seconds.

```
FastIron(config)#ip multicast mcache-age 180
```

Syntax: [no] ip multicast mcache-age <num>

Enabling or Disabling Error and Warning Messages

The device prints error or warning messages when it runs out of software resources or when it receives packets with the wrong checksum or groups. These messages are rate-limited. You can turn off these messages by entering a command such as the following:

```
FastIron(config)#ip multicast verbose-off
```

Syntax: [no] ip multicast verbose-off

Enabling or Disabling PIM Sparse Snooping

PIM snooping should be used only in topologies where multiple PIM sparse routers connect through a device. PIM snooping does not work on a PIM dense mode router which does not send join messages, and traffic to PIM dense ports is stopped. A PIM snooping-enabled device displays a warning if it receives PIM dense join/prune messages. Configure PIM sparse snooping by entering a command such as the following:

```
FastIron(config)#ip pimsm-snooping
```

NOTE: The device must be in passive mode before it can be configured for PIM snooping.

Syntax: [no] ip pimsm-snooping

Configuring the IGMP Mode for a VLAN

You can configure a VLAN to use the active or passive IGMP mode. The default mode is passive. The setting specified for the VLAN overrides the global setting.

- Active - An active IGMP mode device actively sends out IGMP queries to identify multicast groups on the network, and makes entries in the IGMP table based on the group membership reports received.
- Passive - A passive IGMP mode device forwards reports to the router ports which receive queries. IGMP snooping in the passive mode forwards queries to the entire VLAN, but it does not send queries.

To set the IGMP mode for VLAN 20 to active, enter the following commands:

```
FastIron(config)#vlan 20
FastIron(config-vlan-20)#multicast active
```

Syntax: [no] multicast active | passive

Disabling IGMP Snooping for the VLAN

When IGMP snooping is enabled globally, you can still disable it for a specific VLAN. For example, the following commands cause IGMP snooping to be disabled for VLAN 20. This setting overrides the global setting.

```
FastIron(config)#vlan 20
FastIron(config-vlan-20)#multicast disable-multicast-snoop
```

Syntax: [no] multicast disable-multicast-snoop

Enabling PIM Sparse Mode Snooping for the VLAN

You can enable PIM snooping for a specific VLAN. For example, the following commands enable PIM snooping on VLAN 20.

```
FastIron(config)#vlan 20
FastIron(config-vlan-20)#multicast pimsm-snooping
```

Syntax: [no] multicast pimsm-snooping

Disabling PIM Sparse Mode Snooping for the VLAN

When PIM snooping is enabled globally, you can still disable it for a specific VLAN. For example, the following commands disable PIM snooping for VLAN 20. This setting overrides the global setting.

```
FastIron(config)#vlan 20
FastIron(config-vlan-20)#multicast disable-pimsm-snoop
```

Syntax: [no] multicast disable-pimsm-snoop

Configuring the IGMP Version for the VLAN

You can specify the IGMP version for a VLAN. For example, the following commands configure VLAN 20 to use IGMPv3:

```
FastIron(config)#vlan 20
FastIron(config-vlan-20)#multicast version 3
```

Syntax: [no] multicast version 2 | 3

If no IGMP version is specified, then the globally-configured IGMP version is used. If an IGMP version is specified for individual ports in the VLAN, those ports use that version, instead of the VLAN version.

Configuring the IGMP Version for Individual Ports in the VLAN

You can specify the IGMP version for individual ports in a VLAN. For example, the following commands configure ports 0/1/4, 0/1/5, 0/1/6 and 0/2/1 to use IGMPv3. The other ports in the VLAN either use the IGMP version specified with the multicast version command, or the globally-configured IGMP version.

```
FastIron(config)#vlan 20
FastIron(config-vlan-20)#multicast port-version 3 ethe 0/2/1 ethe 0/1/4 to 0/1/6
```

Syntax: [no] multicast port-version 2 | 3 <port-numbers>

Configuring Static Groups to the Entire VLAN or to Specific Ports

A snooping-enabled VLAN cannot forward multicast traffic to ports that do not receive IGMP membership reports. If clients cannot send reports, you can configure a static group which applies to the entire VLAN or only to specific ports. The static group allows packets to be forwarded to the static group ports even though they have no client membership reports. The static group to the entire VLAN is used in VLAN flooding, which consumes less hardware resource than the static group to ports.

The static group drop option discards data traffic to a group in hardware. The group can be any multicast group including groups in the reserved range of 224.0.0.X. The drop option does not apply to IGMP packets, which are always trapped to CPU when snooping is enabled. The drop option applies to the entire VLAN, and cannot be configured for a port list. When the drop option is not specified, the group must exist outside the reserved range.

```
FastIron(config)#vlan 20
FastIron(config-vlan-20)#multicast static-group 224.1.1.1 count 2 ethe 0/1/3 ethe 0/
1/5 to 0/1/7
FastIron(config-vlan-20)#multicast static-group 239.1.1.1 count 3 drop
FastIron(config-vlan-20)#multicast static-group 239.1.1.1
```

Syntax: [no] multicast static-group <ipv4-address> [count <num>] [<port-numbers> | drop]

The ipv4-address parameter is the address of the multicast group.

The count is optional, which allows a contiguous range of groups. Omitting the count <num> is equivalent to the count being 1.

If no <port-numbers> are entered, the static groups apply to the entire VLAN.

Configuring Static Router Ports

The FGS or FLS device forwards all multicast control and data packets to router ports which receive queries. Although router ports are learned, you can force multicast traffic to specified ports even though these ports never receive queries. To configure static router ports, enter commands such as the following:

```
FastIron(config)#vlan 70
FastIron(config-vlan-70)#multicast router-port e 0/1/4 to 0/1/5 e 0/1/8
```

Syntax: [no] multicast router-port <port-numbers>

Turning Off Static Group Proxy

If a device has been configured for static groups, it acts as a proxy and sends membership reports for the static groups when it receives general or group-specific queries. When a static group configuration is removed, it is deleted from active group table immediately. However, leave messages are not sent to the querier, and the querier should age the group out. Proxy activity can be turned off. The default is on. To turn proxy activity off for VLAN 20, enter commands similar to the following:

```
FastIron(config)#vlan 20
FastIron(config-vlan-20)#multicast proxy-off
```

Syntax: [no] multicast proxy-off

Enabling IGMPv3 Membership Tracking and Fast Leave for the VLAN

IGMPv3 gives clients membership tracking and fast leave capability. In IGMPv2, only one client on an interface needs to respond to a router's queries. This can leave some clients invisible to the router, making it impossible to track the membership of all clients in a group. When a client leaves the group, the device sends group-specific queries to the interface to see if other clients on that interface need the data stream of the client who is leaving. If no client responds, the device waits a few seconds before it stops the traffic. You can configure the wait time using the **ip multicast leave-wait-time** command.

IGMPv3 requires every client to respond to queries, allowing the device to track all clients. When tracking is enabled, and an IGMPv3 client sends a leave message and there is no other client, the device immediately stops forwarding traffic to the interface. This feature requires the entire VLAN be configured for IGMPv3 with no IGMPv2 clients. If a client does not send a report during the specified group membership time (the default is 140 seconds), that client is removed from the tracking list.

Every group on a physical port keeps its own tracking record. However, it can only track group membership; it cannot track by (source, group). For example, Client A and Client B belong to group1 but each receives traffic streams from different sources. Client A receives a stream from (source_1, group1) and Client B receives a stream from (source_2, group1). The device still waits for the configured leave-wait-time before it stops the traffic because these two clients are in the same group. If the clients are in different groups, then the waiting period is not applied and traffic is stopped immediately.

To enable the tracking and fast leave feature for VLAN 20, enter the following commands:

```
FastIron(config)#vlan 20
FastIron(config-vlan-20)#multicast tracking
```

Syntax: [no] multicast tracking

The membership tracking and fast leave features are supported for IGMPv3 only. If any port or any client is not configured for IGMPv3, then the multicast tracking command is ignored.

Configuring Fast Leave for IGMPv2

When a device receives an IGMPv2 leave message, it sends out multiple group-specific queries. If no other client replies within the waiting period, the device stops forwarding traffic. When fast-leave-v2 is configured, when the device receives a leave message, it immediately stops forwarding to that port. The device does not send group specific-queries. You must ensure that no snooping-enabled ports have multiple clients. When two devices connect together, the querier should not be configured for fast-leave-v2, since the port might have multiple clients through the non-querier. The number of queries, and the waiting period (in seconds) can be configured using the **ip multicast leave-wait-time** command. The default is 2 seconds. To configure fast leave for IGMPv2, enter the following commands:

```
FastIron(config)#vlan 20
FastIron(config-vlan-20)#multicast fast-leave-v2
```

Syntax: [no] multicast fast-leave-v2

Enabling Fast Convergence

In addition to sending periodic general queries, an active device sends general queries when it detects a new port. However, because the device does not recognize the other device's port up event, multicast traffic might still require up to the query-interval time to resume after a topology change. Fast convergence allows the device to listen to topology change events in L2 protocols such as spanning tree, and then send general queries to shorten the convergence time.

If the L2 protocol cannot detect a topology change, fast convergence may not work in some cases. For example, if the direct connection between two devices switches from one interface to another, the rapid spanning tree protocol (802.1w) considers this optimization, rather than a topology change. In this example, other devices will not receive topology change notifications, and will be unable to send queries to speed up the convergence. Fast convergence works well with the regular spanning tree protocol in this case.

To enable fast-convergency, enter the following commands:

```
FastIron(config)#vlan 70
FastIron(config-vlan-70)#multicast fast-convergence
```

Syntax: multicast fast-convergence

Displaying IGMP Snooping Information

You can display the following IGMP snooping information:

- IGMP information
- Information about VLANs
- Group and forwarding information for VLANs
- PIM sparse snooping information
- IGMP memory pool usage
- Status of IGMP traffic
- IGMP information by VLAN

Displaying IGMP Information

To display information about possible IGMP errors, enter the following command:

```
FastIron#show ip multicast error
snoop SW processed pkt: 173, up-time 160 sec
```

Syntax: show ip multicast error

The following table describes the output from the **show ip multicast error** command:

This Field	Displays
SW processed pkt	The number of multicast packets processed by IGMP snooping.
up-time	The time since the IGMP snooping is enabled.

Displaying IGMP Group Information

To display information about IGMP groups, enter the following command:

```
FastIron#show ip multicast group
p-:physical, ST:static, QR:querier, EX:exclude, IN:include, Y:yes, N:no
VL70 : 3 groups, 4 group-port, tracking_enabled
      group            p-port   ST     QR     life mode    source
1     224.1.1.2        0/1/33   no     yes    120  EX      0
2     224.1.1.1        0/1/33   no     yes    120  EX      0
3     226.1.1.1        0/1/35   yes    yes    100  EX      0
4     226.1.1.1        0/1/33   yes    yes    100  EX      0
```

In this example, an IGMPv2 group is in EXCLUDE mode with a source of 0. The group only excludes traffic from the 0 (zero) source list, which actually means that all traffic sources are included.

To display detailed IGMP group information, enter the following command:

```
FastIron#show ip multicast group 226.1.1.1 detail
Display group 226.1.1.1 in all interfaces in details.
p-:physical, ST:static, QR:querier, EX:exclude, IN:include, Y:yes, N:no
VL70 : 1 groups, 2 group-port, tracking_enabled
      group            p-port   ST     QR     life mode    source
1     226.1.1.1        0/1/35   yes    yes    120  EX      0
    group: 226.1.1.1, EX, permit 0 (source, life):
      life=120, deny 0:
      group            p-port   ST     QR     life mode    source
2     226.1.1.1        0/1/33   yes    yes    120  EX      0
    group: 226.1.1.1, EX, permit 0 (source, life):
      life=120, deny 0:
```

If the tracking and fast leave features are enabled, you can display the list of clients that belong to a particular group by entering the following command:

```
FastIron#show ip multicast group 224.1.1.1 tracking
Display group 224.1.1.1 in all interfaces with tracking enabled.
p-:physical, ST:static, QR:querier, EX:exclude, IN:include, Y:yes, N:no
VL70 : 1 groups, 1 group-port, tracking_enabled
      group            p-port   ST     QR     life mode    source
*** Note: has 1 static groups to the entire vlan, not displayed here
1     224.1.1.1        0/1/33   no     yes    100  EX      0
    receive reports from 1 clients: (age)
      (2.2.100.2 60)
```

Syntax: show ip multicast group [<group-address> [detail] [tracking]]

If you want a report for a specific multicast group, enter that group's address for <group-address>.

Enter detail to display the source list of a specific VLAN.

Enter tracking for information on interfaces that have tracking enabled.

The following table describes the information displayed by the **show ip multicast group** command:

This Field...	Displays...
group	The address of the group (destination address in this case, 224.1.1.1)
p-port	The physical port on which the group membership was received.
ST	**Yes** indicates that the IGMP group was configured as a static group; **No** means the address was learned from reports.
QR	**Yes** means the port is a querier port; No means it is not. A port becomes a non-querier port when it receives a query from a source with a lower source IP address than the device.
life	The number of seconds the group can remain in EXCLUDE mode. An EXCLUDE mode changes to INCLUDE mode if it does not receive an "IS_EX" or "TO_EX" message during a certain period of time. The default is 140 seconds. There is no life displayed in INCLUDE mode.
mode	Indicates current mode of the interface: INCLUDE or EXCLUDE. If the interface is in INCLUDE mode, it admits traffic only from the source list. If an interface is in EXCLUDE mode, it denies traffic from the source list and accepts the rest.
source	Identifies the source list that will be included or excluded on the interface. For example, if an IGMPv2 group is in EXCLUDE mode with a source of 0, the group excludes traffic from the 0 (zero) source list, which actually means that all traffic sources are included.

Displaying IGMP Snooping Mcache Information

The IGMP snooping mcache contains multicast forwarding information for VLANs. To display information in the multicast forwarding mcache, enter the following command:

```
FastIron#show ip multicast mcache
Example: (S G) cnt=: cnt: SW proc. count
    OIF: 0/1/22 TR(0/1/32,0/1/33), TR is trunk, 0/1/32 primary, 0/1/33 output
vlan 1, 1 caches. use 1 VIDX
1    (1.2.10.102 225.1.1.1) cnt=46
     OIF: 0/1/4
     age=0m up-time=45m vidx=4130 (ref-cnt=1)
vlan 70, 1 caches. use 1 VIDX
1    (* 226.1.2.3) cnt=69
     OIF: 0/1/14
     age=0m up-time=59m vidx=4129 (ref-cnt=1)
```

Syntax: show ip multicast mcache

The following table describes the output of the **show ip multicast mcache** command:

This Field...	Displays...
(source group)	Source and group addresses of this data stream. (* group) means match group only; (source group) means match both.
cnt	The number of packets processed in software. Packets are switched in hardware in FGS software release 03.0.00, which increases this number slowly.
OIF	The output interfaces. If `entire vlan` is displayed, this indicates that static groups apply to the entire VLAN.
age	The mcache age. The mcache will be reset to 0 if traffic continues to arrive, otherwise the mcache will be aged out when it reaches the time defined by the **ip multicast mcache-age** command.
uptime	The up time of this mcache in minutes.
vidx	Vidx specifies output port list index. Range is from 4096 to 8191
ref-cnt	The vidx is shared among mcaches having the same output interfaces. Ref-cnt indicates the number of mcaches using this vidx.

Displaying PIM Sparse Snooping Information

PIM sparse mode snooping allows a device to listen for join/prune messages exchanged between PIM routers, which helps reduce unwanted traffic. To display PIM snooping information, enter the following command:

```
FastIron#show ip multicast pimsm-snooping
vlan 1, has 1 caches.
1    (1.2.10.102 225.1.1.1) has 0 pim join ports out of 1 OIF
vlan 70, has 1 caches.
1    (* 226.1.2.3) has 2 pim join ports out of 2 OIF
     0/1/14 (age=60), 0/1/13 (age=60),
     0/1/14 has 1 src: 1.1.30.99(60)
     0/1/13 has 1 src: 1.1.30.99(60)
```

This output shows the number of OIF due to PIM out of the total OIF. The join/prune messages are source-specific. In this case, If the mcache is in (* G), the display function will also print the traffic source information.

Displaying Software Resource Usage for VLANs

To display information about the software resources used, enter the following command:

```
FastIron#show ip multicast resource
                      alloc in-use  avail get-fail    limit  get-mem  size init
igmp group              256      1    255        0    32000        1    16  256
igmp phy port          1024      1   1023        0   200000        1    22 1024
…. entries deleted …
snoop mcache entry      128      2    126        0     8192        3    56  128
total pool memory 109056 bytes
has total 2 forwarding hash
VIDX sharing hash   : size=2      anchor=997  2nd-hash=no  fast-trav=no
Available vidx: 4060. IGMP/MLD use 2
```

Syntax: show ip multicast resource

The following table describes the output from the **show ip multicast resource** command:

This Field...	Displays...
alloc	The allocated number of units.
in-use	The number of units which are currently being used.
avail	The number of available units.
get-fail	This displays the number of resource failures. NOTE: It is important to pay attention to this field.
limit	The upper limit of this expandable field. The limit of `multicast group` is configured by the **system-max igmp-max-group-addr** command. The limit of `snoop mcache entry` is configured by the **system-max multicast-snoop-mcache** command.
get-mem	The number of memory allocation. This number should continue to increase.
size	The size of a unit (in bytes).
init	The initial allocated amount of memory. More memory may be allocated if resources run out.
Available vidx	The output interface (OIF) port mask used by mcache. The entire device has a maximum of 4096 vidx. Different mcaches with the same OIF share the same vidx. If vidx is not available, the stream cannot be hardware-switched.

Displaying Status of IGMP Snooping Traffic

To display status information for IGMP snooping traffic, enter the following command:

```
FastIron#show ip multicast traffic
IGMP snooping: Total Recv: 22, Xmit: 26
Q: query, Qry: general Q,  G-Qry: group Q,  GSQry: group-source Q, Mbr: member
Recv      QryV2     QryV3     G-Qry     GSQry     MbrV2     MbrV3     Leave
VL1           0         0         0         0         4         0         0
VL70         18         0         0         0         0         0         0
Recv       IsIN      IsEX      ToIN      ToEX     ALLOW     BLOCK   Pkt-Err
VL1           0         4         0         0         0         0         0
VL70          0         0         0         0         0         0         0

Send      QryV2     QryV3     G-Qry     GSQry     MbrV2     MbrV3
VL1           0         0         8         0         0         0
VL70          0         0         0         0         0        18
VL70   pimsm-snooping, Hello:    12,  Join/Prune:     9
```

Syntax: show ip multicast traffic

The following table describes the information displayed by the **show ip multicast traffic** command.

This Field...	Displays...
Q	Query
Qry	General Query
QryV2	Number of general IGMPv2 queries received or sent.
QryV3	Number of general IGMPv3 queries received or sent.
G-Qry	Number of group-specific queries received or sent.
GSQry	Number of group source-specific queries received or sent.
Mbr	The membership report.
MbrV2	The IGMPv2 membership report.
MbrV3	The IGMPv3 membership report.
IsIN	Number of source addresses that were included in the traffic.
IsEX	Number of source addresses that were excluded in the traffic.
ToIN	Number of times the interface mode changed from EXCLUDE to INCLUDE.
ToEX	Number of times the interface mode changed from INCLUDE to EXCLUDE.
ALLO	Number of times that additional source addresses were allowed on the interface.
BLK	Number of times that sources were removed from an interface.
Pkt-Err	Number of packets having errors, such as checksum.
Pimsm-snooping hello, join, prune	Number of PIM sparse hello, join, and prune packets

Displaying IGMP Snooping Information by VLAN

You can display IGMP snooping information for all VLANs or for a specific VLAN. For example, to display IGMP snooping information for VLAN 70, enter the following command:

```
FastIron#show ip multicast vlan 70
version=2, query-t=30, group-aging-t=140, max-resp-t=3, other-qr-present-t=63
VL70: dft V2, vlan cfg passive, , pimsm (vlan cfg), track, 0 grp, 1 (*G) cache,
rtr ports,
    router ports: 0/1/13(140) 1.1.70.3, 0/1/20(180) 1.1.70.2, 0/1/14(180)
  0/1/13  has    0 groups, non-QR (passive), default V2
  0/1/14  has    0 groups, non-QR (passive), default V2
  0/1/20  has    0 groups, non-QR (passive), default V2
```

Syntax: show ip multicast vlan [<vlan-id>]

If you do not specify a <vlan-id>, information for all VLANs is displayed.

The following table describes the information displayed by the **show ip multicast vlan** command.

This Field...	Displays...
version	The IGMP version number
query-t	How often a querier sends a general query on the interface.
group-aging-t	The number of seconds membership groups can be members of this group before aging out.
rtr-port	The router ports which are the ports receiving queries. The display `router ports: 0/1/13(140) 1.1.70.3` means port 0/1/13 has a querier with 1.1.70.3 address, and a remaining life of 140 seconds.
max-resp-t	The maximum number of seconds a client waits before it replies to the query.
non-QR	Indicates that the port is a non-querier.
QR	Indicates that the port is a querier.

Clear IGMP Snooping Commands

The clear IGMP snooping commands should be used only in troubleshooting conditions, or to recover from errors.

Clear IGMP Counters on VLANs

To clear IGMP snooping on error and traffic counters for all VLANs, enter the following command:

```
FastIron#clear ip multicast counters
```

Syntax: clear ip multicast counters

Clear IGMP mcache

To clear the mcache on all VLANs, enter the following command:

```
FastIron#clear ip multicast mcache
```

Syntax: clear ip multicast mcache

Clear mcache on a Specific VLAN

To clear the mcache on a specific VLAN, enter the following command:

```
FastIron#clear ip multicast vlan 10 mcache
```

Syntax: clear ip multicast vlan <vlan-id> mcache

The <vlan-id> parameter specifies the specific VLAN to clear the cache.

Clear Traffic on a Specific VLAN

To clear the traffic counters on a specific VLAN, enter the following command:

```
FastIron#clear ip multicast vlan 10 traffic
```

Syntax: clear ip multicast vlan <vlan-id> traffic

The <vlan-id> parameter specifies the specific VLAN on which to clear the traffic counters

Chapter 27

Configuring IP Multicast Traffic Reduction for the FastIron X Series Switch

This chapter describes how to configure the following IP multicast traffic reduction parameters on a Foundry FastIron X Series switch:

- Internet Group Management Protocol (IGMP) snooping
- Protocol Independent Multicast Sparse Mode (PIM SM) traffic snooping

NOTE: This chapter applies to FastIron X Series devices only. For information about configuring IP multicast traffic reduction on the FastIron GS and FastIron LS, see “Configuring IP Multicast Traffic Reduction for the FastIron GS and FastIron LS” on page 26-1.

IGMP Snooping Overview

When a device processes a multicast packet, by default, it broadcasts the packets to all ports except the incoming port of a VLAN. Packets are flooded by hardware without going to the CPU. This behavior causes some clients to receive unwanted traffic.

IGMP snooping provides multicast containment by forwarding traffic to only the ports that have IGMP receivers for a specific multicast group (destination address). A device maintains the IGMP group membership information by processing the IGMP reports and leave messages, so traffic can be forwarded to ports receiving IGMP reports.

An IPv4 multicast address is a destination address in the range of 224.0.0.0 to 239.255.255.255. Addresses of 224.0.0.X are reserved. Because packets destined for these addresses may require VLAN flooding, devices do not do snooping in the reserved range. Data packets destined to addresses in the reserved range are flooded to the entire VLAN by hardware, and mirrored to the CPU. Multicast data packets destined for the non-reserved range of addresses are snooped. A client must send IGMP reports in order to receive traffic. If an application outside the reserved range requires VLAN flooding, the user must configure a static group that applies to the entire VLAN. In addition, a static group with the drop option can discard multicast data packets to a specified group in hardware, including addresses in the reserved range.

An IGMP device's responsibility is to broadcast general queries periodically, and to send group queries when receiving a leave message, to confirm that none of the clients on the port still want specific traffic before removing the traffic from the port. IGMP V2 lets clients specify what group (destination address) will receive the traffic but not to specify the source of the traffic. IGMP V3 is for source-specific multicast traffic, adding the capability for clients to INCLUDE or EXCLUDE specific traffic sources. An IGMP V3 device port state could be INCLUDE or EXCLUDE, and there are different types of group records for client reports.

The interfaces respond to general or group queries by sending a membership report that contains one or more of the following records associated with a specific group:

- Current-state record that indicates from which sources the interface wants to receive and not receive traffic.

This record contains the source address of interfaces and whether or not traffic will be included (IS_IN) or not excluded (IS_EX) from this source.

- Filter-mode-change record. If the interface state changes from IS_IN to IS_EX, a TO_EX record is included in the membership report. Likewise, if the interface state changes from IS_EX to IS_IN, a TO_IN record appears in the membership report.
- An IGMP V2 leave report is equivalent to a TO_IN (empty) record in IGMP V3. This record means that no traffic from this group will be received regardless of the source.
- An IGMP V2 group report is equivalent to an IS_EX (empty) record in IGMP V3. This record means that all traffic from this group will be received regardless of source.
- Source-list-change record. If the interface wants to add or remove traffic sources from its membership report, the report can contain an ALLOW record, which includes a list of new sources from which the interface wishes to receive traffic. It can also contain a BLOCK record, which lists the current traffic sources from which the interface wants to stop receiving traffic.

IGMP protocols provide a method for clients and a device to exchange messages, and let the device build a database indicating which port wants what traffic. The protocols do not specify forwarding methods. They require IGMP snooping or multicast protocols such as PIM or DVMRP to handle packet forwarding. PIM and DVMRP can route multicast packets within and outside a VLAN, while IGMP snooping can switch packets only within a VLAN.

If a VLAN is not IGMP snooping-enabled, it floods multicast data and control packets to the entire VLAN in hardware. When snooping is enabled, IGMP packets are trapped to the CPU. Data packets are mirrored to the CPU in addition to being VLAN flooded. The CPU then installs hardware resources, so that subsequent data packets can be switched to desired ports in hardware without going to the CPU. If there is no client report or port to queriers for a data stream, the hardware resource drops it.

MAC-Based Implementation

On both switch and router software images, IGMP snooping is MAC-based. This differs from IGMP snooping on the BigIron/FastIron router images, which match on both IP source and group (S,G) entries programmed in the Layer 4 CAM. In contrast, the FastIron X Series images match on Layer 2 destination MAC address entries (*,G).

When Layer 2 CAM is used, traffic is switched solely based on the destination MAC address. Consequently, traffic of the same group coming to the same port, regardless of its source, is switched in the same way. In addition, the lowest 23 bits of the group address are mapped to a MAC address. In this way, multiple groups (for example, 224.1.1.1 and 225.1.1.1) have the same MAC address. Groups having the same MAC address are switched to the same destination ports, which are the superset of individual group output ports. Thus, the use of Layer 2 CAM might cause unwanted packets to be sent to some ports. However, the switch generally needs far less Layer 2 CAM than it does Layer 4 CAM, which is required for each stream with a different source and group.

IGMP V1, V2, and V3 Snooping Support

Table 27.1 shows IGMP snooping version support by software release on FastIron X Series devices.

Table 27.1: IGMP Snooping Support

Software Release	IGMP Version Support	Supported in Software Code...
01.1.00 – 02.1.01	IGMP V1 snooping IGMP V2 snooping	L2
02.2.00 – 04.0.01	IGMP V1 snooping IGMP V2 snooping	L2 BL3 L3

Table 27.1: IGMP Snooping Support (Continued)

04.1.00 and later releases	IGMP V1 snooping	L2
	IGMP V2 snooping	BL3
	IGMP V3 snooping	L3

Queriers and Non-Queriers

An IGMP snooping-enabled Foundry device can be configured as a querier (active) or non-querier (passive). An IGMP querier sends queries; a non-querier listens for IGMP queries and forwards them to the entire VLAN. VLANs can be independently configured to be queriers or non-queriers. If a VLAN has a connection to a PIM/ DVMRP-enabled port on another router, the VLAN should be configured as a non-querier. When multiple IGMP snooping devices are connected together, and there is no connection to a PIM/DVMRP-enabled port, one of the devices should be configured as a querier. If multiple devices are configured as queriers, after these devices exchange queries, then all except the winner stop sending queries. The device with the lowest address becomes the querier. Although the system will work when multiple devices are configured as queriers, Foundry recommends that only one device (preferably the one with the traffic source) is configured as a querier.

The non-queriers always forward multicast data traffic and IGMP messages to router ports which receive IGMP queries or PIM/DVMRP hellos. Foundry recommends that you configure the device with the data traffic source (server) as a querier. If a server is attached to a non-querier, the non-querier always forwards traffic to the querier regardless of whether there are any clients on the querier.

NOTE: In a topology of one or more connecting devices, at least one device must be running PIM/DVMRP, or configured as active. Otherwise, none of the devices can send out queries, and traffic cannot be forwarded to clients.

IGMP Snooping Enhancements in Software Release 04.1.00

This section describes the enhancements to IGMP snooping in software release 04.1.00

Support for IGMP V3 Snooping

See “IGMP Snooping Overview” on page 27-1.

VLAN-Specific Configuration

IGMP snooping can be enabled on some VLANs or on all VLANs. Each VLAN can be independently configured to be a quierier or non-quierier and can be configured for IGMP V2 or IGMP V3. In general, the **ip multicast** commands apply globally to all VLANs except those configured with VLAN-specific **multicast** commands. The VLAN-specific multicast commands supersede the global **ip multicast** commands.

IGMP snooping can be configured for IGMP V2 or IGMP V3 on individual ports of a VLAN. An interface or router sends the queries and reports that include its IGMP version specified on it. The version configuration only applies to sending queries. The snooping device recognizes and processes IGMP V2 and IGMP V3 packets regardless of the version configuration.

To avoid version deadlock, an interface retains its version configuration even when it receives a report with a lower version.

Tracking and Fast Leave

FastIron X Series devices support fast leave for IGMP V2, and tracking and fast leave for IGMP V3. Fast leave stops the traffic immediately when the port receives a leave message. Tracking traces all IGMP V3 clients. See “Enabling IGMP V3 Membership Tracking and Fast Leave for the VLAN” on page 27-13 and “Enabling Fast Leave for IGMP V2” on page 27-14.

IGMP Snooping and Layer 3 Multicast Routing Together on the Same Device

Software release FSX 04.1.00 adds support for global Layer 2 IP multicast traffic reduction (IGMP snooping) and Layer 3 multicast routing (DVMRP/PIM-Sparse/PIM-Dense) together on the same device in the full Layer 3 software image, as long as the Layer 2 feature configuration is at the VLAN level. Releases prior to FSX 04.1.00 support global Layer 2 IP multicast traffic reduction and Layer 3 multicast routing. However, they were mutually exclusive and configuring IGMP snooping on a VLAN together with DVMRP/PIM-Sparse/PIM-Dense was not allowed.

See “IP Multicast Protocols and IGMP Snooping on the Same Device” on page 30-57.

Configuration Notes and Feature Limitations

- Layer 2 IGMP snooping is automatically enabled with Layer 3 multicast routing. If Layer 3 multicast routing is enabled on your system, do not attempt to enable Layer 2 IGMP snooping.
- The default IGMP version is V2.
- A user can configure the maximum numbers of group address entries.
- An IGMP device can be configured to rate-limit the forwarding IGMP V2 membership reports to queriers.
- The device supports static groups. The device acts as a proxy to send IGMP reports for the static groups when receiving queries.
- A user can configure static router ports to force all multicast traffic to these specific ports.
- If a VLAN has a connection to a PIM/DVMRP-enabled port on another router, the VLAN should be configured as a non-querier (passive). When multiple snooping devices connect together and there is no connection to PIM/DVMRP ports, one device should be configured as a querier (active). If multiple devices are configured as active (queriers), only one will keep sending queries after exchanging queries.
- The querier must configure an IP address to send out queries.
- IGMP snooping requires hardware resource. Hardware resource is installed only when there is data traffic. If resource is inadequate, the data stream without a resource is mirrored to the CPU in addition to being VLAN flooded, which can cause high CPU usage. Foundry recommends that you avoid global enabling of snooping unless necessary.
- IGMP snooping requires clients to send membership reports in order to receive data traffic. If a client application does not send reports, you must configure static groups on the snooping VLAN to force traffic to client ports. Note that servers (traffic sources) are not required to send IGMP memberships.

PIM SM Traffic Snooping Overview

When multiple PIM sparse routers connect through a snooping-enabled device, the Foundry device always forwards multicast traffic to these routers. For example, PIM sparse routers R1, R2 and R3 connect through a device. Assume R2 needs traffic, and R1 sends it to the device, which forwards it to both R2 and R3, even though R3 does not need it. A PIM SM snooping-enabled device listens to join and prune messages exchanged by PIM sparse routers, and stops traffic to the router that sends prune messages. This allows the device to forward the data stream to R2 only.

PIM SM traffic snooping requires IGMP snooping to be enabled on the device. IGMP snooping configures the device to listen for IGMP messages. PIM SM traffic snooping provides a finer level of multicast traffic control by configuring the device to listen specifically for PIM SM join and prune messages sent from one PIM SM router to another through the device.

PIM SM Snooping Support

Table 27.2 shows PIM SM snooping version support by software release on FastIron X Series devices.

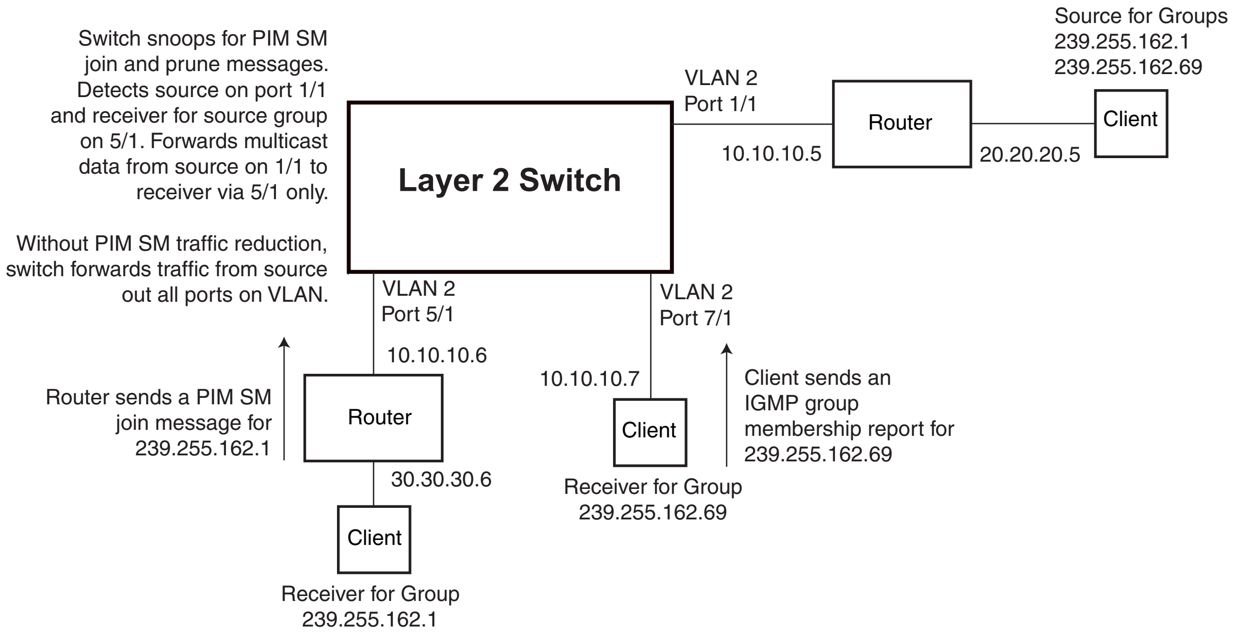
Table 27.2: IGMP Snooping Support

Software Release	Version Support	Supported in Software Code...
02.2.00 to 04.0.01*x*	PIM SM V2 snooping	L2
04.1.00 and later	PIM SM V2 snooping	L2 BL3 L3

Application Examples

Figure 27.1 shows an example application of the PIM SM traffic snooping feature. In this example, a device is connected through an IP router to a PIM SM group source that is sending traffic for two PIM SM groups. The device also is connected to a receiver for each of the groups.

Figure 27.1 PIM SM Traffic Reduction in an Enterprise Network


Switch snoops for PIM SM join and prune messages. Detects source on port 1/1 and receiver for source group on 5/1. Forwards multicast data from source on 1/1 to receiver via 5/1 only.
Without PIM SM traffic reduction, switch forwards traffic from source out all ports on VLAN.
Layer 2 Switch
VLAN 2
Port 1/1
10.10.10.5
Router
20.20.20.5
Client
Source for Groups
239.255.162.1
239.255.162.69
VLAN 2
Port 5/1
10.10.10.6
Router
30.30.30.6
Client
Receiver for Group
239.255.162.1
Router sends a PIM SM join message for 239.255.162.1
VLAN 2
Port 7/1
10.10.10.7
Client
Receiver for Group
239.255.162.69
Client sends an IGMP group membership report for 239.255.162.69

When PIM SM traffic snooping is enabled, the device starts listening for PIM SM join and prune messages and IGMP group membership reports. Until the device receives a PIM SM join message or an IGMP group membership report, the device forwards IP multicast traffic out all ports. Once the device receives a join message or group membership report for a group, the device forwards subsequent traffic for that group only on the ports from which the join messages or IGMP reports were received.

In this example, the router connected to the receiver for group 239.255.162.1 sends a join message toward the group's source. Since PIM SM traffic snooping is enabled on the device, the device examines the join message to learn the group ID, then makes a forwarding entry for the group ID and the port connected to the receiver's router. The next time the device receives traffic for 239.255.162.1 from the group's source, the device forwards the traffic only on port 5/1, since that is the only port connected to a receiver for the group.

Notice that the receiver for group 239.255.162.69 is directly connected to the device. As result, the device does not see a join message on behalf of the client. However, since IGMP snooping also is enabled, the device uses the IGMP group membership report from the client to select the port for forwarding traffic to group 239.255.162.69 receivers.

The IGMP snooping feature and the PIM SM traffic snooping feature together build a list of groups and forwarding ports for the VLAN. The list includes PIM SM groups learned through join messages as well as MAC addresses learned through IGMP group membership reports. In this case, even though the device never sees a join message for the receiver for group 239.255.162.69, the device nonetheless learns about the receiver and forwards group traffic to the receiver.

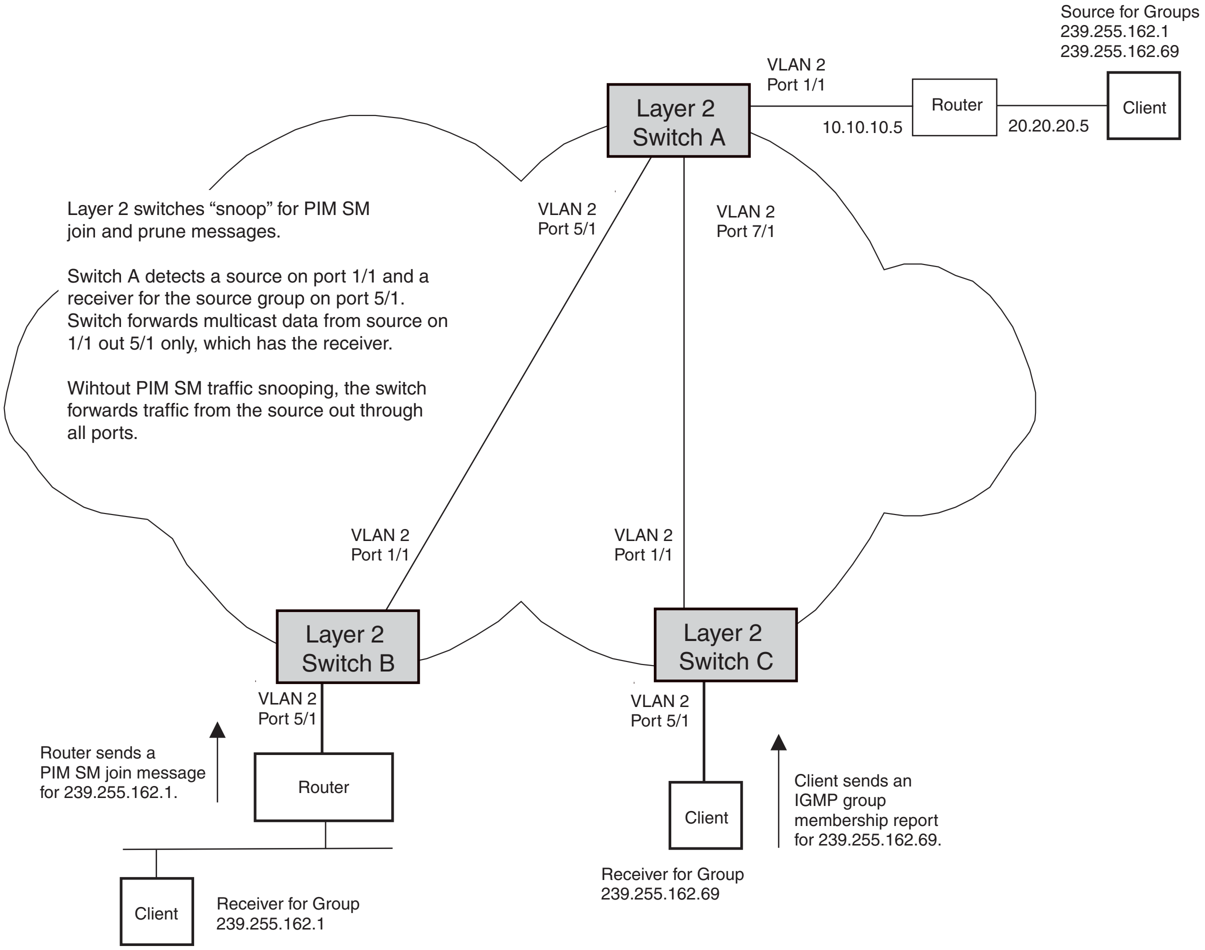
The device stops forwarding IP multicast traffic on a port for a group if the port receives a prune message for the group.

Notice that the ports connected to the source and the receivers are all in the same port-based VLAN on the device. This is required for the PIM SM snooping feature. The devices on the edge of the Global Ethernet cloud are configured for IGMP snooping and PIM SM traffic snooping. Although this application uses multiple devices, the feature has the same requirements and works the same way as it does on a single device.

The following figure shows another example application for PIM SM traffic snooping. This example shows devices on the edge of a Global Ethernet cloud (a Layer 2 Packet over SONET cloud). Assume that each device is attached to numerous other devices such as other Layer 2 Switches and Layer 3 Switches (routers).

NOTE: This example assumes that the devices are actually FastIron devices running Layer 2 Switch software.

Figure 27.2 PIM SM Traffic Reduction in Global Ethernet Environment



The devices on the edge of the Global Ethernet cloud are configured for IGMP snooping and PIM SM traffic snooping. Although this application uses multiple devices, the feature has the same requirements and works the same way as it does on a single device.

Configuration Notes and Limitations

- PIM SM snooping applies only to PIM SM version 2 (PIM SM V2).
- Starting with software release 04.1.00, PIM SM traffic snooping is supported in the Layer 2, base Layer 3, and full Layer 3 code. Releases prior to 04.1.00 support PIM SM traffic snooping in the Layer 2 code only.
- IGMP snooping must be enabled on the device that will be running PIM SM snooping. The PIM SM traffic snooping feature requires IGMP snooping.

  NOTE: Use the passive mode of IGMP snooping instead of the active mode. The passive mode assumes that a router is sending group membership queries as well as join and prune messages on behalf of receivers. The active mode configures the device to send group membership queries.

- All the device ports connected to the source and receivers or routers must be in the same port-based VLAN.

- The PIM SM snooping feature assumes that the group source and the device are in different subnets and communicate through a router. The source must be in a different IP subnet than the receivers. A PIM SM router sends PIM join and prune messages on behalf of a multicast group receiver only when the router and the source are in different subnet. When the receiver and source are in the same subnet, they do not need the router in order to find one another. They find one another directly within the subnet.

  The device forwards all IP multicast traffic by default. Once you enable IGMP snooping and PIM SM traffic snooping, the device initially blocks all PIM SM traffic instead of forwarding it. The device forwards PIM SM traffic to a receiver only when the device receives a join message from the receiver. Consequently, if the source and the downstream router are in the same subnet, and PIM SM traffic snooping is enabled, the device blocks the PIM SM traffic and never starts forwarding the traffic. This is because the device never receives a join message from the downstream router for the group. The downstream router and group find each other without a join message because they are in the same subnet.

NOTE: If the “route-only” feature is enabled on a Layer 3 Switch, PIM SM traffic snooping will not be supported.

Configuring IGMP Snooping

Configuring IGMP snooping on a Foundry device consists of the following global, VLAN-specific, and port-specific tasks:

Global Tasks

- “Configuring the IGMP V3 Snooping Software Resource Limits”
- “Enabling IGMP Snooping Globally on the Device”
- “Configuring the Global IGMP Mode”
- “Configuring the Global IGMP Version”
- “Modifying the Age Interval for Group Membership Entries”
- “Modifying the Query Interval (Active IGMP Snooping Mode Only)”
- “Modifying the Maximum Response Time”
- “Configuring Report Control” (rate limiting)
- “Modifying the Wait Time Before Stopping Traffic When Receiving a Leave Message”
- “Modifying the Multicast Cache Age Time”
- “Enabling or Disabling Error and Warning Messages”

VLAN-Specific Tasks

- “Configuring the IGMP Mode for a VLAN” (active or passive)
- “Disabling IGMP Snooping on a VLAN”
- “Configuring the IGMP Version for a VLAN”
- “Configuring Static Router Ports”
- “Turning Off Static Group Proxy”
- “Enabling IGMP V3 Membership Tracking and Fast Leave for the VLAN”
- “Enabling Fast Leave for IGMP V2”
- “Enabling Fast Convergence”

Port-Specific Tasks

- “Disabling Transmission and Receipt of IGMP Packets on a Port”
- “Configuring the IGMP Version for Individual Ports in a VLAN”

Configuring the IGMP V3 Snooping Software Resource Limits

By default, the system supports up to 512 IGMP snooping multicast cache (mcache) entries and a maximum of 8K IGMP group addresses. If necessary, you can change the default values using the procedures in this section.

About IGMP Snooping Mcache Entries and Group Addresses

An IGMP snooping group address entry is created when an IGMP join message is received for a group. An IGMP snooping mcache entry is created when data traffic is received for that group. Each mcache entry represents one data stream, and multiple mcache entries (up to 32) can share the same hardware (MAC) address entry. The egress port list for the mcache entry is obtained from the IGMP group address entry. If there is no existing IGMP group address entry when an mcache entry is created, data traffic for that multicast group is dropped in hardware. If there is an existing IGMP group address entry when an mcache is created, data traffic for that multicast group is switched in hardware.

Changing the Maximum Number of Supported IGMP Snooping Mcache Entries

Platform Support: FastIron X Series devices running software release 04.1.00 or later – L2, BL3, L3

When IGMP snooping is enabled, by default, the system supports up to 512 IGMP snooping mcache entries. If necessary, you can change the maximum number of IGMP snooping cache entries supported on the device. To do so, enter a command such as the following:

```
FastIron(config)#system-max igmp-snoop-mcache 2000
```

Syntax: [no] system-max igmp-snoop-mcache <num>

where <num> is a value between 256 and 8192. The default is 512.

Setting the Maximum Number of IGMP Group Addresses

Platform Support: FastIron X Series devices running software release 04.1.00 or later – L2, BL3, L3

When IGMP snooping is enabled, by default, the system supports up to 8K of IGMP group addresses. The configurable range is from 256 to 32768. The configured number is the upper limit of an expandable database. Client memberships exceeding the group limit are not processed. Enter a command such as the following to define the maximum number of IGMP group addresses:

```
FastIron(config)#system-max igmp-max-group-addr 1600
```

Syntax: [no] system-max igmp-max-group-addr <num>

where <num> is a value between 256 and 32768. The default is 8192.

Enabling IGMP Snooping Globally on the Device

Use the procedures in this section to enable IGMP snooping on a global basis.

Configuration Notes

- Layer 2 IGMP snooping is automatically enabled with Layer 3 multicast routing. If Layer 3 multicast routing is enabled on your system, do not attempt to enable Layer 2 IGMP snooping.
- If the "route-only" feature is enabled on the Layer 3 Switch, then IP multicast traffic reduction will not be supported.
- IGMP snooping is not supported on the default VLAN of Layer 3 Switches.

Releases Prior to 04.1.00

Use the following command to globally enable IGMP snooping in releases prior to 04.1.00:

```
FastIron(config)#ip multicast
```

The above command enables IGMP V2. Starting with software release 04.1.00, IGMP V3 is also supported. See “Releases 04.1.00 and Later” .

Syntax: [no] ip multicast

Releases 04.1.00 and Later

Starting in release 04.1.00, when you globally enable IGMP snooping, you can specify IGMP V2 or IGMP V3 for the device. The following command causes the device to use IGMP V3:

```
FastIron(config)#ip multicast version 3
```

Syntax: [no] ip multicast version 2|3

If you do not specify a version number, IGMP V2 is assumed.

Configuring the IGMP Mode

You can configure active or passive IGMP modes on the Foundry device. The default mode is passive. If you specify an IGMP mode for a VLAN, it overrides the global setting.

- **Active** - When active IGMP mode is enabled, a Foundry device actively sends out IGMP queries to identify multicast groups on the network, and makes entries in the IGMP table based on the group membership reports it receives.

  NOTE: Routers in the network generally handle this operation. Use the active IGMP mode only when the device is in a stand-alone Layer 2 Switched network with no external IP multicast router attachments. In this case, enable the active IGMP mode on only one of the devices and leave the other devices configured for passive IGMP mode.

- **Passive** - When passive IGMP mode is enabled, it forwards reports to the router ports which receive queries. IGMP snooping in the passive mode does not send queries. However, it forwards queries to the entire VLAN.

Configuring the Global IGMP Mode

To globally set the IGMP mode to active, enter the following command:

```
FastIron(config)#ip multicast active
```

Syntax: [no] ip multicast [active | passive]

If you do not enter either *active* or *passive*, the passive mode is assumed.

Configuring the IGMP Mode for a VLAN

Platform Support: FastIron X Series devices running software release 04.1.00 and later – L2, BL3, L3.

If you specify an IGMP mode for a VLAN, it overrides the global setting.

To set the IGMP mode for VLAN 20 to active, enter the following commands:

```
FastIron(config)#vlan 20
FastIron(config-vlan-20)#multicast active
```

Syntax: [no] multicast active | passive

Configuring the IGMP Version

Platform Support: FastIron X Series devices running software release 04.1.00 and later – L2, BL3, L3

Use the commands in this section to configure the IGMP version in releases 04.1.00 and later.

Configuring the Global IGMP Version

You can globally specify IGMP V2 or IGMP V3 for the device. The following command causes the device to use IGMP V3:

```
FastIron(config)#ip multicast version 3
```

Syntax: [no] ip multicast version 2|3

Configuring the IGMP Version for a VLAN

You can specify the IGMP version for a VLAN. For example, the following commands configure VLAN 20 to use IGMP V3:

```
FastIron(config)#vlan 20
FastIron(config-vlan-20)#multicast version 3
```

Syntax: [no] multicast version 2 | 3

If no IGMP version is specified, then the globally-configured IGMP version is used. If an IGMP version is specified for individual ports in the VLAN, those ports use that version, instead of the VLAN version.

Configuring the IGMP Version for Individual Ports in a VLAN

You can specify the IGMP version for individual ports in a VLAN. For example, the following commands configure ports 4, 5, and 6 to use IGMP V3. The other ports in the VLAN either use the IGMP version specified with the multicast version command, or the globally-configured IGMP version.

```
FastIron(config)#vlan 20
FastIron(config-vlan-20)#multicast port-version 3 ethe 2/4 to 2/6
```

Syntax: [no] multicast port-version 2 | 3 <port-numbers>

Disabling IGMP Snooping on a VLAN

Platform Support: FastIron X Series devices running software release 04.1.00 and later – L2, BL3, L3

When IGMP snooping is enabled globally, you can still disable it for a specific VLAN. For example, the following commands cause IGMP snooping to be disabled for VLAN 20. This setting overrides the global setting.

```
FastIron(config)#vlan 20

FastIron(config-vlan-20)#multicast disable-multicast-snoop
```

Syntax: [no] multicast disable-multicast-snoop

Disabling Transmission and Receipt of IGMP Packets on a Port

When a VLAN is snooping-enabled, all IGMP packets are trapped to the CPU without hardware VLAN flooding. The CPU can block IGMP packets to and from a multicast-disabled port, and does not add it to the output interfaces of hardware resources. This prevents the disabled port from receiving multicast traffic. However, if static groups to the entire VLAN are defined, the traffic from these groups is VLAN flooded, including to disabled ports. Traffic from disabled ports cannot be blocked in hardware, and is switched in the same way as traffic from enabled ports.

This command has no effect on a VLAN that is not snooping-enabled because all multicast traffic is VLAN flooded.

To disable transmission and receipt of IGMP packets on a port, enter commands such as the following:

```
FastIron(config)#interface ethernet 3
FastIron(config-if-e1000-3)#ip-multicast-disable
```

The above commands disable IGMP snooping on port 1/5 but does not affect the state of IGMP on other ports.

Syntax: [no] ip-multicast-disable

Modifying the Age Interval for Group Membership Entries

When the device receives a group membership report, it makes an entry for that group in the IGMP group table. The age interval specifies how long the entry can remain in the table before the device receives another group membership report. When multiple devices connect together, all devices should be configured for the same age interval, which should be at least twice the length of the query interval, so that missing one report won't stop traffic. Non-querier age intervals should be the same as the age interval of the querier.

To modify the age interval, enter a command such as the following:

```
FastIron(config)#ip multicast age-interval 280
```

Syntax: [no] ip multicast age-interval <interval>

The <interval> parameter specifies the aging time. You can specify a value from 20 - 7200 seconds. The default is 260 seconds.

Modifying the Query Interval (Active IGMP Snooping Mode Only)

If IP multicast traffic reduction is set to active mode, you can modify the query interval to specify how often the device sends group membership queries. When multiple queriers connect together, they should all be configured with the same query interval.

To modify the query interval, enter a command such as the following:

```
FastIron(config)#ip multicast query-interval 120
```

Syntax: [no] ip multicast query-interval <interval>

The <interval> parameter specifies the time between queries. You can specify a value from 10 - 3600 seconds. The default is 125 seconds.

Modifying the Maximum Response Time

The maximum response time is the number of seconds that a client can wait before responding to a query sent by the switch. The default response time is 10 seconds maximum.

To change the maximum response time, enter a command such as the following

```
FastIron(config)#ip multicast max-response-time 5
```

Syntax: [no] ip multicast max-response-time <interval>

For <interval>, enter a value from 1 – 10 seconds. The default is 10 seconds.

Configuring Report Control

A device in passive mode forwards reports and leave messages from clients to the upstream router ports that are receiving queries.

You can configure report control to rate-limit report forwarding within the same group to no more than once every 10 seconds. This rate-limiting does not apply to the first report answering a group-specific query.

NOTE: This feature applies to IGMP V2 only. The leave messages are not rate limited.

IGMP V2 membership reports of the same group from different clients are considered to be the same and are rate-limited.

Use the following command to alleviate report storms from many clients answering the upstream router query.

```
 FastIron(config)#ip multicast report-control
```

Syntax: [no] ip multicast report-control

The original command, **ip igmp-report-control,** has been renamed to **ip multicast report-control**. The original command is still accepted; however, it is renamed when you issue a **show configuration** command.

Modifying the Wait Time Before Stopping Traffic When Receiving a Leave Message

You can define the wait time before stopping traffic to a port when a leave message is received. The device sends group-specific queries once per second to ask if any client in the same port still needs this group. The value range is from 1 to 5, and the default is 2. Due to internal timer granularity, the actual wait time is between n and (n+1) seconds (n is the configured value).

```
FastIron(config)#ip multicast leave-wait-time 1
```

Syntax: [no] ip multicast leave-wait-time <num>

<num> is the number of seconds from 1 to 5. The default is 2 seconds.

Modifying the Multicast Cache Age Time

You can set the time for an mcache to age out when it does not receive traffic. The traffic is hardware switched. One minute before aging out an mcache, the device mirrors a packet of this mcache to CPU to reset the age. If no data traffic arrives within one minute, this mcache is deleted. A lower value quickly removes resources consumed by idle streams, but it mirrors packets to CPU often. A higher value is recommended only data streams are continually arriving.

```
FastIron(config)#ip multicast mcache-age 180
```

Syntax: [no] ip multicast mcache-age <num>

<num> is the number of seconds from 60 to 3600. The default is 60 seconds.

Enabling or Disabling Error and Warning Messages

The device prints error or warning messages when it runs out of software resources or when it receives packets with the wrong checksum or groups. These messages are rate-limited. You can turn off these messages by entering a command such as the following:

```
FastIron(config)#ip multicast verbose-off
```

Syntax: [no] ip multicast verbose-off

Configuring Static Router Ports

The Foundry device forwards all multicast control and data packets to router ports which receive queries. Although router ports are learned, you can force multicast traffic to specified ports even though these ports never receive queries. To configure static router ports, enter commands such as the following:

```
FastIron(config)#vlan 70
FastIron(config-vlan-70)#multicast router-port e 4 to 5 e 8
```

Syntax: [no] multicast router-port <port-numbers>

Turning Off Static Group Proxy

If a device has been configured for static groups, it acts as a proxy and sends membership reports for the static groups when it receives general or group-specific queries. When a static group configuration is removed, it is deleted from the active group table immediately. However, leave messages are not sent to the querier, and the querier should age out the group. Proxy activity can be turned off. The default is on. To turn proxy activity off for VLAN 20, enter commands similar to the following:

```
FastIron(config)#vlan 20
FastIron(config-vlan-20)#multicast proxy-off
```

Syntax: [no] multicast proxy-off

Enabling IGMP V3 Membership Tracking and Fast Leave for the VLAN

Platform Support: FastIron X Series devices running software release 04.1.00 and later – L2, BL3, L3

IGMP V3 gives clients membership tracking and fast leave capability. In IGMP V2, only one client on an interface needs to respond to a router's queries. This can leave some clients invisible to the router, making it impossible to track the membership of all clients in a group. When a client leaves the group, the device sends group-specific queries to the interface to see if other clients on that interface need the data stream of the client who is leaving. If no client responds, the device waits a few seconds before it stops the traffic. You can configure the wait time using the **ip multicast leave-wait-time** command.

IGMP V3 requires every client to respond to queries, allowing the device to track all clients. When tracking is enabled, and an IGMP V3 client sends a leave message and there is no other client, the device immediately stops forwarding traffic to the interface. This feature requires the entire VLAN be configured for IGMP V3 with no IGMP

V2 clients. If a client does not send a report during the specified group membership time (the default is 260 seconds), that client is removed from the tracking list.

Every group on a physical port keeps its own tracking record. However, it can only track group membership; it cannot track by (source, group). For example, Client A and Client B belong to group1 but each receives traffic streams from different sources. Client A receives a stream from (source_1, group1) and Client B receives a stream from (source_2, group1). The device still waits for the configured leave-wait-time before it stops the traffic because these two clients are in the same group. If the clients are in different groups, then the waiting period is not applied and traffic is stopped immediately.

To enable the tracking and fast leave feature for VLAN 20, enter the following commands:

```
FastIron(config)#vlan 20
FastIron(config-vlan-20)#multicast tracking
```

Syntax: [no] multicast tracking

The membership tracking and fast leave features are supported for IGMP V3 only. If any port or any client is not configured for IGMP V3, then the multicast tracking command is ignored.

Enabling Fast Leave for IGMP V2

Platform Support: FastIron X Series devices running software release 04.1.00 and later – L2, BL3, L3

When a device receives an IGMP V2 leave message, it sends out multiple group-specific queries. If no other client replies within the waiting period, the device stops forwarding traffic. When fast-leave-v2 is configured, when the device receives a leave message, it immediately stops forwarding to that port. The device does not send group specific-queries. You must ensure that no snooping-enabled ports have multiple clients. When two devices connect together, the querier should not be configured for fast-leave-v2, since the port might have multiple clients through the non-querier. The number of queries, and the waiting period (in seconds) can be configured using the **ip multicast leave-wait-time** command. The default is 2 seconds. To configure fast leave for IGMP V2, enter the following commands:

```
FastIron(config)#vlan 20
FastIron(config-vlan-20)#multicast fast-leave-v2
```

Syntax: [no] multicast fast-leave-v2

Enabling Fast Convergence

Platform Support: FastIron X Series devices running software release 04.1.00 and later – L2, BL3, L3

In addition to sending periodic general queries, an active device sends general queries when it detects a new port. However, because the device does not recognize the other device's port up event, multicast traffic might still require up to the query-interval time to resume after a topology change. Fast convergence allows the device to listen to topology change events in Layer 2 protocols such as spanning tree, and then send general queries to shorten the convergence time.

If the Layer 2 protocol cannot detect a topology change, fast convergence may not work in some cases. For example, if the direct connection between two devices switches from one interface to another, the rapid spanning tree protocol (802.1w) considers this optimization, rather than a topology change. In this example, other devices will not receive topology change notifications, and will be unable to send queries to speed up the convergence. Fast convergence works well with the regular spanning tree protocol in this case.

To enable fast-convergency, enter the following commands:

```
FastIron(config)#vlan 70
FastIron(config-vlan-70)#multicast fast-convergence
```

Syntax: multicast fast-convergence

Configuring PIM SM Snooping

Configuring PIM SM snooping on a Foundry device consists of the following global and VLAN-specific tasks:

Global Tasks

- “Enabling or Disabling PIM SM Snooping”

VLAN-Specific Tasks

- “Enabling PIM SM Snooping on a VLAN”
- “Disabling PIM SM Snooping on a VLAN”

Enabling or Disabling PIM SM Snooping

PIM SM snooping should be used only in topologies where multiple PIM sparse routers connect through a device. PIM SM snooping does not work on a PIM dense mode router which does not send join messages, and traffic to PIM dense ports is stopped. A PIM SM snooping-enabled device displays a warning if it receives PIM dense join/prune messages.

To enable PIM sparse snooping globally, enter a command such as the following:

```
FastIron(config)#ip pimsm-snooping
```

This command enables PIM SM traffic snooping. The PIM SM traffic snooping feature assumes that the network has routers that are running PIM SM.

NOTE: The device must be in passive mode before it can be configured for PIM SM snooping.

To disable the feature, enter the following command:

```
FastIron(config)#no ip pimsm-snooping
```

If you also want to disable IP multicast traffic reduction, enter the following command:

```
FastIron(config)#no ip multicast
```

Syntax: [no] ip pimsm-snooping

Enabling PIM SM Snooping on a VLAN

You can enable PIM SM snooping for a specific VLAN. For example, the following commands enable PIM SM snooping on VLAN 20.

```
FastIron(config)#vlan 20
FastIron(config-vlan-20)#multicast pimsm-snooping
```

Syntax: [no] multicast pimsm-snooping

Disabling PIM SM Snooping on a VLAN

When PIM SM snooping is enabled globally, you can still disable it for a specific VLAN. For example, the following commands disable PIM SM snooping for VLAN 20. This setting overrides the global setting.

```
FastIron(config)#vlan 20
FastIron(config-vlan-20)#multicast disable-pimsm-snoop
```

Syntax: [no] multicast disable-pimsm-snoop

IGMP Snooping Show Commands

This section shows how to display information about IGMP snooping, including:

- “Displaying the IGMP Snooping Configuration”
- “Displaying IGMP Snooping Errors”
- “Displaying IGMP Group Information”
- “Displaying IGMP Snooping Mcache Information”

- “Displaying Usage of Hardware Resource by Multicast Groups”
- “Displaying Software Resource Usage for VLANs”
- “Displaying the Status of IGMP Snooping Traffic”

Displaying the IGMP Snooping Configuration

To display the global IGMP snooping configuration, enter the following command at any level of the CLI:

```
FastIron#show ip multicast
Summary of all vlans. Please use "sh ip mu vlan <vlan-id>" for details
Version=2, Intervals: Query=125, Group Age=260, Max Resp=10, Other Qr=260
VL10: cfg V3, vlan cfg passive, , pimsm (vlan cfg), 1 grp, 0 (SG) cache, no rtr port
```

To display the IGMP snooping information for a specific VLAN (release 04.1.00 or later), enter a command such as the following:

```
FastIron#show ip multicast vlan 10
Version=3, Intervals: Query=10, Group Age=260, Max Resp=10, Other Qr=30
VL10: cfg V3, vlan cfg passive, , pimsm (vlan cfg), 3 grp, 1 (SG) cache, no rtr
port,
  e2      has    3 groups, non-QR (passive), default V3
  **** Warning! has V2 client (life=240),
    group: 239.0.0.3, life = 240
    group: 224.1.1.2, life = 240
    group: 224.1.1.1, life = 240

  e4      has    0 groups, non-QR (passive), default V3
```

Syntax: show ip multicast vlan [<vlan-id>]

If you do not specify a <vlan-id>, information for all VLANs is displayed.

This display shows the following information.

This Field...	Displays...
Version	The global IGMP version. In this example, the device is configured for IGMP version 2.
Query	How often a querier sends a general query on the interface. In this example, the general queries are sent every 125 seconds.
Group Age	The number of seconds membership goups can be members of this group before aging out.
Max Resp	The maximum number of seconds a client waits before replying to a query.
Other Qr	How long it took a switch with a lower IP address to become a new querier. This value is 2 x Query + Max Resp.
cfg	The IGMP version for the specified VLAN. In this example, **VL10: cfg V3** indicates that VLAN 10 is configured for IGMP V3.

This Field...	Displays...
vlan cfg	The IGMP configuration mode, which is either **passive** or **active**.
pimsm	Indicates that PIM SM is enabled on the VLAN.
rtr port	The router ports, which are the ports receiving queries.

Displaying IGMP Snooping Errors

To display information about possible IGMP errors, enter the following command:

```
FastIron#show ip multicast error
snoop SW processed pkt: 173, up-time 160 sec
```

Syntax: show ip multicast error

The following table describes the output from the **show ip multicast error** command:

This Field	Displays
SW processed pkt	The number of multicast packets processed by IGMP snooping.
up-time	The time since the IGMP snooping is enabled.

Displaying IGMP Group Information

Platform Support: FastIron X Series devices running software release 04.1.00 or later – L2, BL3, L3

To display information about IGMP groups, enter the following command:

```
FastIron#show ip multicast group
p-:physical, ST:static, QR:querier, EX:exclude, IN:include, Y:yes, N:no
VL70 : 3 groups, 4 group-port, tracking_enabled
      group            p-port    ST     QR     life mode    source
1     224.1.1.2        1/33      no     yes    120  EX      0
2     224.1.1.1        1/33      no     yes    120  EX      0
3     226.1.1.1        1/35      yes    yes    100  EX      0
4     226.1.1.1        1/33      yes    yes    100  EX      0
```

In this example, an IGMP V2 group is in EXCLUDE mode with a source of 0. The group only excludes traffic from the 0 (zero) source list, which actually means that all traffic sources are included.

To display detailed IGMP group information for a specific group, enter the following command:

```
FastIron#show ip multicast group 226.1.1.1 detail
Display group 226.1.1.1 in all interfaces in details.
p-:physical, ST:static, QR:querier, EX:exclude, IN:include, Y:yes, N:no
VL70 : 1 groups, 2 group-port, tracking_enabled
      group           p-port   ST     QR     life mode    source
1     226.1.1.1       1/35     yes    yes    120  EX      0
    group: 226.1.1.1, EX, permit 0 (source, life):
      life=120, deny 0:
      group           p-port   ST     QR     life mode    source
2     226.1.1.1       1/33     yes    yes    120  EX      0
    group: 226.1.1.1, EX, permit 0 (source, life):
      life=120, deny 0:
```

If the tracking and fast leave features are enabled, you can display the list of clients that belong to a particular group by entering the following command:

```
FastIron#show ip multicast group 224.1.1.1 tracking
Display group 224.1.1.1 in all interfaces with tracking enabled.
p-:physical, ST:static, QR:querier, EX:exclude, IN:include, Y:yes, N:no
VL70 : 1 groups, 1 group-port, tracking_enabled
      group           p-port   ST     QR     life mode    source
*** Note: has 1 static groups to the entire vlan, not displayed here
1     224.1.1.1       1/33     no     yes    100  EX      0
    receive reports from 1 clients: (age)
      (2.2.100.2 60)
```

Syntax: show ip multicast group [<group-address> [detail] [tracking]]

If you want a report for a specific multicast group, enter that group's address for <group-address>.

Enter detail to display the source list of a specific VLAN.

Enter tracking for information on interfaces that have tracking enabled.

The following table describes the information displayed by the **show ip multicast group** command:

This Field...	Displays...
group	The address of the group (destination address in this case, 224.1.1.1)
p-port	The physical port on which the group membership was received.
ST	**Yes** indicates that the IGMP group was configured as a static group; **No** means the address was learned from reports.
QR	**Yes** means the port is a querier port; No means it is not. A port becomes a non-querier port when it receives a query from a source with a lower source IP address than the device.
life	The number of seconds the group can remain in EXCLUDE mode. An EXCLUDE mode changes to INCLUDE mode if it does not receive an "IS_EX" or "TO_EX" message during a certain period of time. The default is 260 seconds. There is no life displayed in INCLUDE mode.

This Field...	Displays...
mode	Indicates current mode of the interface: INCLUDE or EXCLUDE. If the interface is in INCLUDE mode, it admits traffic only from the source list. If an interface is in EXCLUDE mode, it denies traffic from the source list and accepts the rest.
source	Identifies the source list that will be included or excluded on the interface. For example, if an IGMP V2 group is in EXCLUDE mode with a source of 0, the group excludes traffic from the 0 (zero) source list, which actually means that all traffic sources are included.

Displaying IGMP Snooping Mcache Information

Platform Support: FastIron X Series devices running software release 04.1.00 or later – L2, BL3, L3

The IGMP snooping mcache contains multicast forwarding information for VLANs. To display information in the multicast forwarding mcache, enter the following command:

```
FastIron#show ip multicast mcache
Example: (S G) cnt=: cnt is number of SW processed packets
    OIF: e1/22 TR(1/32,1/33), TR is trunk, e1/32 primary, e1/33 output
vlan 10, 1 caches. use 1 VIDX
1    (10.10.10.2 239.0.0.3) cnt=0
     OIF: tag e2
     age=2s up-time=2s change=2s vidx=8191 (ref-cnt=1)
```

Syntax: show ip multicast mcache

The following table describes the output of the **show ip multicast mcache** command:

This Field...	Displays...
(source group)	Source and group addresses of this data stream. (* group) means match group only; (source group) means match both.
cnt	The number of packets processed in software. Packets are switched in hardware, which increases this number slowly.
OIF	The output interfaces. If `entire vlan` is displayed, this indicates that static groups apply to the entire VLAN.
age	The mcache age. The mcache will be reset to 0 if traffic continues to arrive, otherwise the mcache will be aged out when it reaches the time defined by the **ip multicast mcache-age** command.
uptime	The up time of this mcache in seconds.
vidx	Vidx specifies output port list index. Range is from 4096 to 8191
ref-cnt	The vidx is shared among mcaches having the same output interfaces. Ref-cnt indicates the number of mcaches using this vidx.

Displaying Usage of Hardware Resource by Multicast Groups

You can display how much hardware resource (CAM and FID) is currently being used by multicast groups by entering a command such as the following at any level of the CLI:

```
FastIron#show ip multicast hardware
Hw resource is shared by groups with the same lower 23 bits
VLAN ID 2
Total number of HW resource in vlan: 1
1     Group: 0.1.1.1, HW-ref-cnt=1, vidx 8191
   Forwarding Port: 2
VLAN ID 1
Total number of HW resource in vlan: 0
```

If you want to display the amount of hardware resource that is currently being used by a specific group, enter a command such as the following at any level of the CLI:

```
FastIron#show ip multicast hardware 239.255.163.2
VLAN ID 100
 Group: 239.255.163.2, HW-ref-cnt=1, fid 08a9, cam 10, dma=8,
   Forwarding Port: 1 2
group 239.255.163.2 in 1 vlans
```

Syntax: show ip multicast hardware [<group-address> | vlan <vlan-id>]

Enter the address of a group for <group-address> if you want to display the hardware resource usage of a particular group.

Likewise, enter the ID of a VLAN for <vlan-id> if you want display the hardware resource usage of groups in a VLAN.

The display shows the following information:

This Field...	Displays...
VLAN ID	The port-based VLAN to which the information listed below applies.
Total number of HW resource in VLAN	The number of resources in the VLAN.
Group	Address of the IP multicast group that is using the entry. In the display above, group “0.1.1.1” is using this entry. The field HW-ref-cnt shows the number of groups that are sharing this entry. Multiple groups could share one entry because only low 23 bits are significant. **Note**: The vidx, fid, cam, and dma values are used by Foundry Technical Support for troubleshooting.
Forwarding Port	The forwarding ports for the IP multicast group.

Displaying Software Resource Usage for VLANs

Platform Support: FastIron X Series devices running software release 04.1.00 or later – L2, BL3, L3

To display information about the software resources used, enter the following command:

```
FastIron#show ip multicast resource
                      alloc in-use  avail get-fail    limit  get-mem  size init
igmp group              256      1    255        0    32000        1    16  256
igmp phy port          1024      1   1023        0   200000        1    22 1024
…. entries deleted …
snoop mcache entry      128      2    126        0     8192        3    56  128
total pool memory 109056 bytes
has total 2 forwarding hash
VIDX sharing hash   : size=2      anchor=997  2nd-hash=no  fast-trav=no
Available vidx: 4060. IGMP/MLD use 2
```

Syntax: show ip multicast resource

The following table describes the output from the **show ip multicast resource** command:

This Field...	Displays...
alloc	The allocated number of units.
in-use	The number of units which are currently being used.
avail	The number of available units.
get-fail	This displays the number of resource failures. NOTE: It is important to pay attention to this field.
limit	The upper limit of this expandable field. The limit of `multicast group` is configured by the **system-max igmp-max-group-addr** command. The limit of `snoop mcache entry` is configured by the **system-max multicast-snoop-mcache** command.
get-mem	The number of memory allocation. This number should continue to increase.
size	The size of a unit (in bytes).
init	The initial allocated amount of memory. More memory may be allocated if resources run out.
Available vidx	The output interface (OIF) port mask used by mcache. The entire device has a maximum of 4096 vidx. Different mcaches with the same OIF share the same vidx. If vidx is not available, the stream cannot be hardware-switched.

Displaying the Status of IGMP Snooping Traffic

Platform Support: FastIron X Series devices running software release 04.1.00 or later – L2, BL3, L3

To display status information for IGMP snooping traffic, enter the following command:

```
FastIron#show ip multicast traffic
IGMP snooping: Total Recv: 22, Xmit: 26
Q: query, Qry: general Q,  G-Qry: group Q,  GSQry: group-source Q, Mbr: member
Recv      QryV2     QryV3     G-Qry     GSQry     MbrV2     MbrV3     Leave
VL1           0         0         0         0         4         0         0
VL70         18         0         0         0         0         0         0
Recv       IsIN      IsEX      ToIN      ToEX     ALLOW     BLOCK   Pkt-Err
VL1           0         4         0         0         0         0         0
VL70          0         0         0         0         0         0         0

Send      QryV2     QryV3     G-Qry     GSQry     MbrV2     MbrV3
VL1           0         0         8         0         0         0
VL70          0         0         0         0         0        18
VL70   pimsm-snooping, Hello:    12,  Join/Prune:     9
```

Syntax: show ip multicast traffic

The following table describes the information displayed by the **show ip multicast traffic** command.

This Field...	Displays...
Q	Query
Qry	General Query
QryV2	Number of general IGMP V2 queries received or sent.
QryV3	Number of general IGMP V3 queries received or sent.
G-Qry	Number of group-specific queries received or sent.
GSQry	Number of group source-specific queries received or sent.
Mbr	The membership report.
MbrV2	The IGMP V2 membership report.
MbrV3	The IGMP V3 membership report.
IsIN	Number of source addresses that were included in the traffic.
IsEX	Number of source addresses that were excluded in the traffic.
ToIN	Number of times the interface mode changed from EXCLUDE to INCLUDE.
ToEX	Number of times the interface mode changed from INCLUDE to EXCLUDE.
ALLO	Number of times that additional source addresses were allowed on the interface.
BLK	Number of times that sources were removed from an interface.
Pkt-Err	Number of packets having errors, such as checksum.
Pimsm-snooping hello, join, prune	Number of PIM sparse hello, join, and prune packets

PIM SM Snooping Show Commands

This section shows how to display information about PIM SM snooping, including:

- "Displaying PIM SM Snooping Information"
- "Displaying PIM SM Snooping Information on a Layer 2 Switch"
- "Displaying PIM SM Snooping Information for a Specific Group or Source Group Pair"

Displaying PIM SM Snooping Information

To display PIM SM snooping information, enter the following command:

```
FastIron#show ip multicast pimsm-snooping
vlan 1, has 2 caches.
1    (* 230.1.1.1) has 1 pim join ports out of 1 OIF
     1 (age=60)
     1 has 1 src: 20.20.20.66(60)
2    (* 230.2.2.2) has 1 pim join ports out of 1 OIF
     1 (age=60)
     1 has 1 src: 20.20.20.66(60)
```

This output shows the number of PIM join outgoing interfaces (OIF) out of the total OIF. The join/prune messages are source-specific. In this case, If the mcache is in (* G), the display function will also print the traffic source information.

Syntax: show ip multicast pimsm-snooping [<vlan-id>]

Use the <vlan-id> parameter to display PIM SM snooping information for a specific VLAN.

Displaying PIM SM Snooping Information on a Layer 2 Switch

You can display PIM SM snooping information for all groups by entering the following command at any level of the CLI on a Layer 2 Switch:

```
FastIron#show ip multicast pimsm-snooping vlan 100
VLAN ID 100, total 3 entries
PIMSM Neighbor list:
        1.100.100.12     : 3/3 expire 120 s
        1.100.100.10     : 3/2 expire 170 s
        1.100.100.7      : 3/1 expire 160 s
1     Group: 224.0.1.22, fid 08ac, NO cam
   Forwarding Port: 3/3
   PIMv2 Group Port: 3/3
   (Source, Port) list: 1 entries
2     Group: 239.255.162.2, fid 08aa, cam 8
   Forwarding Port: 3/1 3/2
   PIMv2 Group Port: 3/1 3/2
   (Source, Port) list: 3 entries
3     Group: 239.255.163.2, fid 08a9, cam 10
   Forwarding Port: 3/1 3/2
   PIMv2 Group Port: 3/1 3/2
   (Source, Port) list: 3 entries
VLAN ID 4008, total 0 entries
PIMSM Neighbor list:
```

Syntax: show ip pimsm-snooping vlan <vlan-id>

Enter the ID of the VLAN for the **vlan** <vlan-id> parameter.

If you want to display PIM SM snooping information for one source or one group, enter a command as in the following example. The command also displays the (source, port) list of the group.

```
FastIron#show ip pimsm-snooping 239.255.163.2
Show pimsm snooping group 239.255.163.2 in all vlan
VLAN ID 100
 Group: 239.255.163.2, fid 08a9, cam 10
   Forwarding Port: 3/1 3/2
   PIMv2 Group Port: 3/1 3/2
   (Source, Port) list: 3 entries
       1    192.168.176.44, age=0, port: 3/2
       2    158.158.158.158, age=0, port: 3/1
       3    1.1.7.1, age=0, port: 3/2
```

Syntax: show ip pimsm-snooping <group-address> | <source-address>

If the address you entered is within the range of source addresses, then the router treats it as the source address. Likewise, if the address falls in the range of group addresses, then the router assumes that you are requesting a report for that group.

This display shows the following information.

This Field...	Displays...
VLAN ID	The port-based VLAN to which the information listed below apply and the number of members in the VLAN.
PIM SM Neighbor list	The PIM SM routers that are attached to the Layer 2 Switch's ports in the VLAN. The value following "expires" indicates how many seconds the Layer 2 Switch will wait for a hello message from the neighbor before determining that the neighbor is no longer present and removing the neighbor from the list.
Multicast Group	The IP address of the multicast group. **Note**: The fid and camindex values are used by Foundry Technical Support for troubleshooting.
Forwarding Port	The port(s) attached to the group's receivers. A port is listed here when it receives a join message for the group, an IGMP membership report for the group, or both.
PIMv2 Group Port	The port(s) on which the Layer 2 Switch has received PIM SM join messages for the group.
Source, Port list	The IP address of each PIM SM source and the Layer 2 Switch ports connected to the receivers of the source.

Displaying PIM SM Snooping Information for a Specific Group or Source Group Pair

To display PIM SM snooping information for a specific group, enter a command such as the following at any level of the CLI:

```
FastIron#show ip multicast pimsm-snooping 230.1.1.1
Show pimsm snooping group 230.1.1.1 in all vlans
vlan 10,has 2 caches.
1 (*230.1.1.1) has 1 pim join ports out of 1 OIF
   1(age=120)
   1 has 1 src:20.20.20.66(120)
```

To display PIM SM snooping information for a specific (source, group) pair, enter a command such as the following at any level of the CLI:

```
FastIron#show ip multicast pimsm-snooping 230.2.2.2 20.20.20.66
Show pimsm snooping source 20.20.20.66, group 230.2.2.2 in all vlans
vlan 10:(*230.2.2.2) has 1 pim join ports out of 2 OIF
   1(age=0)
   1 has 1 src:20.20.20.66(0)
```

Syntax: show ip multicast pimsm-snooping <group-address> [<source-ip-address>]

The Foundry device determines which address is the group address and which one is the source address based on the ranges that the address fall into. If the address is within the range of source addresses, then the router treats it as the source address. Likewise, if the address falls in the range of group addresses, then the router assumes it is a group address.

The output shows the following information.

This Field...	Displays...
vlan	The VLAN membership ID of the source.
port	The port on which the source is sending traffic. In this example, the port number is 1.
age	The age of the port, in seconds.
src	The source address and age. The age (number of seconds) is indicated in brackets immediately following the source.

Clear Commands for IGMP Snooping

The clear IGMP snooping commands should be used only in troubleshooting conditions, or to recover from errors.

Clearing the IGMP Mcache

Platform Support: FastIron X Series devices running software release 04.1.00 or later – L2, BL3, L3

To clear the mcache on all VLANs, enter the following command:

```
FastIron#clear ip multicast mcache
```

Syntax: clear ip multicast mcache

Clearing the Mcache on a Specific VLAN

Platform Support: FastIron X Series devices running software release 04.1.00 or later – L2, BL3, L3

To clear the mcache on a specific VLAN, enter the following command:

```
FastIron#clear ip multicast vlan 10 mcache
```

Syntax: clear ip multicast vlan <vlan-id> mcache

The <vlan-id> parameter specifies the specific VLAN to clear the cache.

Clearing Traffic on a Specific VLAN

Platform Support: FastIron X Series devices running software release 04.1.00 or later – L2, BL3, L3

To clear the traffic counters on a specific VLAN, enter the following command:

```
FastIron#clear ip multicast vlan 10 traffic
```

Syntax: clear ip multicast vlan <vlan-id> traffic

The <vlan-id> parameter specifies the specific VLAN on which to clear the traffic counters.

Clearing IGMP Counters on VLANs

Platform Support: FastIron X Series devices running software release 04.1.00 or later – L2, BL3, L3

To clear IGMP snooping on error and traffic counters for all VLANs, enter the following command:

```
FastIron#clear ip multicast counters
```

Syntax: clear ip multicast counters

Chapter 28

Configuring IPv6 Connectivity on a FastIron X Series Switch

This chapter explains IPv6 addressing and features and how to configure them on a Foundry FastIron X Series switch.

NOTE: With exception to IPv6 host features, the commands in this chapter are supported on FastIron X Series switches running software release 04.1.00 and later. IPv6 host features, covered in "IPv6 Host Support" on page 28-12, have been supported on FastIron X Series switches since software release 02.4.00.

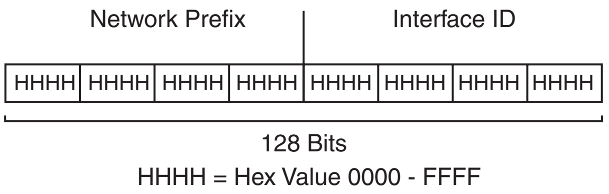
NOTE: This chapter does not describe IPv6 routing protocols, which are covered in separate chapters throughout this guide.

IPv6 Addressing Overview

IPv6 was designed to replace IPv4, the Internet protocol that is most commonly used currently throughout the world. IPv6 increases the number of network address bits from 32 (IPv4) to 128 bits, which provides more than enough unique IP addresses to support all of the network devices on the planet into the future. IPv6 is expected to quickly become the network standard.

An IPv6 address is composed of 8 fields of 16-bit hexadecimal values separated by colons (:). Figure 28.1 shows the IPv6 address format.

Figure 28.1 IPv6 Address Format


Network Prefix
Interface ID
HHHH HHHH HHHH HHHH HHHH HHHH HHHH HHHH
128 Bits
HHHH = Hex Value 0000 - FFFF

As shown in Figure 28.1, HHHH is a 16-bit hexadecimal value, while H is a 4-bit hexadecimal value. The following is an example of an IPv6 address:

2001:0000:0000:0200:002D:D0FF:FE48:4672

Note that this IPv6 address includes hexadecimal fields of zeros. To make the address less cumbersome, you can do the following:

- Omit the leading zeros; for example, 2001:0:0:200:2D:D0FF:FE48:4672.
- Compress the successive groups of zeros at the beginning, middle, or end of an IPv6 address to two colons (::) once per address; for example, 2001::200:2D:D0FF:FE48:4672.

When specifying an IPv6 address in a command syntax, keep the following in mind:

- You can use the two colons (::) only once in the address to represent the longest successive hexadecimal fields of zeros
- The hexadecimal letters in IPv6 addresses are not case-sensitive

As shown in Figure 28.1, the IPv6 network prefix is composed of the left-most bits of the address. As with an IPv4 address, you can specify the IPv6 prefix using the <prefix>/<prefix-length> format, where the following applies:

The <prefix> parameter is specified as 16-bit hexadecimal values separated by a colon.

The <prefix-length> parameter is specified as a decimal value that indicates the left-most bits of the IPv6 address.

The following is an example of an IPv6 prefix:

2001:FF08:49EA:D088::/64

IPv6 Address Types

As with IPv4 addresses, you can assign multiple IPv6 addresses to a switch interface. Table 28.1 presents the three major types of IPv6 addresses that you can assign to a switch interface.

A major difference between IPv4 and IPv6 addresses is that IPv6 addresses support ***scope***, which describes the topology in which the address may be used as a unique identifier for an interface or set of interfaces.

Unicast and multicast addresses support scoping as follows:

- Unicast addresses support two types of scope: global scope and local scope. In turn, local scope supports site-local addresses and link-local addresses. Table 28.1 describes global, site-local, and link-local addresses and the topologies in which they are used.

- Multicast addresses support a scope field, which Table 28.1 describes.

.

Table 28.1: IPv6 address types

Address Type	Description	Address Structure
Unicast	An address for a single interface. A packet sent to a unicast address is delivered to the interface identified by the address.	Depends on the type of the unicast address: • Aggregatable global address—An address equivalent to a global or public IPv4 address. The address structure is as follows: a fixed prefix of 2000::/3 (001), a 45-bit global routing prefix, a 16-bit subnet ID, and a 64-bit interface ID. • Site-local address—An address used within a site or intranet. (This address is similar to a private IPv4 address.) A site consists of multiple network links. The address structure is as follows: a fixed prefix of FEC0::/10 (1111 1110 11), a 16-bit subnet ID, and a 64-bit interface ID. • Link-local address—An address used between directly connected nodes on a single network link. The address structure is as follows: a fixed prefix of FE80::/10 (1111 1110 10) and a 64-bit interface ID. • IPv4-compatible address—An address used in IPv6 transition mechanisms that tunnel IPv6 packets dynamically over IPv4 infrastructures. The address embeds an IPv4 address in the low-order 32 bits and the high-order 96 bits are zeros. The address structure is as follows: 0:0:0:0:0:0:A.B.C.D. • Loopback address—An address (0:0:0:0:0:0:0:1 or ::1) that a switch can use to send an IPv6 packet to itself. You cannot assign a loopback address to a physical interface. • Unspecified address—An address (0:0:0:0:0:0:0:0 or ::) that a node can use until you configure an IPv6 address for it.
Multicast	An address for a set of interfaces belonging to different nodes. Sending a packet to a multicast address results in the delivery of the packet to all interfaces in the set.	A multicast address has a fixed prefix of FF00::/8 (1111 1111). The next 4 bits define the address as a permanent or temporary address. The next 4 bits define the scope of the address (node, link, site, organization, global).
Anycast	An address for a set of interfaces belonging to different nodes. Sending a packet to an anycast address results in the delivery of the packet to the closest interface identified by the address.	An anycast address looks similar to a unicast address, because it is allocated from the unicast address space. If you assign a unicast address to multiple interfaces, it is an anycast address. An interface assigned an anycast address must be configured to recognize the address as an anycast address. An anycast address can be assigned to a switch only. An anycast address must not be used as the source address of an IPv6 packet.

A switch automatically configures a link-local unicast address for an interface by using the prefix of FE80::/10 (1111 1110 10) and a 64-bit interface ID. The 128-bit IPv6 address is then subjected to duplicate address detection to ensure that the address is unique on the link. If desired, you can override this automatically configured address by explicitly configuring an address.

IPv6 Stateless Autoconfiguration

Foundry routers use the IPv6 stateless autoconfiguration feature to enable a host on a local link to automatically configure its interfaces with new and globally unique IPv6 addresses associated with its location. The automatic configuration of a host interface is performed without the use of a server, such as a Dynamic Host Configuration Protocol (DHCP) server, or manual configuration.

The automatic configuration of a host interface works in the following way: a switch on a local link periodically sends switch advertisement messages containing network-type information, such as the 64-bit prefix of the local link and the default route, to all nodes on the link. When a host on the link receives the message, it takes the local link prefix from the message and appends a 64-bit interface ID, thereby automatically configuring its interface. (The 64-bit interface ID is derived from the MAC address of the host's NIC.) The 128-bit IPv6 address is then subjected to duplicate address detection to ensure that the address is unique on the link.

The duplicate address detection feature verifies that a unicast IPv6 address is unique before it is assigned to a host interface by the stateless auto configuration feature. Duplicate address detection uses neighbor solicitation messages to verify that a unicast IPv6 address is unique.

NOTE: For the stateless auto configuration feature to work properly, the advertised prefix length in switch advertisement messages must always be 64 bits.

The IPv6 stateless autoconfiguration feature can also automatically reconfigure a host's interfaces if you change the ISP for the host's network. (The host's interfaces must be renumbered with the IPv6 prefix of the new ISP.)

The renumbering occurs in the following way: a switch on a local link periodically sends advertisements updated with the prefix of the new ISP to all nodes on the link. (The advertisements still contain the prefix of the old ISP.) A host can use the addresses created from the new prefix and the existing addresses created from the old prefix on the link. When you are ready for the host to use the new addresses only, you can configure the lifetime parameters appropriately using the **ipv6 nd prefix-advertisement** command. During this transition, the old prefix is removed from the switch advertisements. At this point, only addresses that contain the new prefix are used on the link..

IPv6 CLI Command Support

Table 28.2 lists the IPv6 CLI commands supported.

Table 28.2: IPv6 CLI Command Support

IPv6 Command	Description	Switch Code	Router Code
clear ipv6 cache	Deletes all entries in the dynamic host cache.		X
clear ipv6 mld-snooping	Deletes MLD-snooping-related counters or cache entries.	X	X
clear ipv6 neighbor	Deletes all dynamic entries in the IPv6 neighbor table.	X	X
clear ipv6 ospf	Clears OSPF-related entries.		X
clear ipv6 rip	Clears RIP-related entries.		X
clear ipv6 route	Deletes all dynamic entries in the IPv6 route table.		X
clear ipv6 traffic	Resets all IPv6 packet counters.	X	X
clear ipv6 tunnel	Clears statistics for IPv6 tunnels	X	X
copy tftp	Downloads a copy of a Foundry software image from a TFTP server into the system flash using IPv6.	X	X

Table 28.2: IPv6 CLI Command Support

IPv6 Command	Description	Switch Code	Router Code
debug ipv6	Displays IPv6 debug information.	X	X
ipv6 access-class	Configures access control for IPv6 management traffic.	X	X
ipv6 access-list	Configures an IPv6 access list for IPv6 access control.	X	X
ipv6 address	Configures an IPv6 address on an interface	X	X
ipv6 debug	Enables IPv6 debugging.	X	X
ipv6 dns domain-name	Configures an IPv6 domain name.	X	X
ipv6 dns server-address	Configures an IPv6 DNS server address.	X	X
ipv6 enable	Enables IPv6 on an interface.	X	X
ipv6 hop-limit	Sets the IPv6 hop limit.		X
ipv6 icmp	Configures IPv6 ICMP parameters		X
Ipv6 load-sharing	Enables IPv6 load sharing		X
Ipv6 mld-snooping	Configures MLD snooping	X	X
ipv6 mtu	Configures the maximum length of an IPv6 packet that can be transmitted on a particular interface.		X
ipv6 nd	Configures neighbor discovery.		X
ipv6 neighbor	Maps a static IPv6 address to a MAC address in the IPv6 neighbor table.		X
ipv6 ospf	Configures OSPF V3 parameters on an interface.		X
ipv6 prefix-list	Builds an IPv6 prefix list.		X
ipv6 redirects	Enables the sending of ICMP redirect messages on an interface.		X
ipv6 rip	Configures RIPng parameters on an interface		X
ipv6 route	Configures an IPv6 static route.		X
ipv6 router	Enables an IPv6 routing protocol.		X
ipv6 traffic-filter	Applies an IPv6 ACL to an interface.	X	X
ipv6 unicast-routing	Enables IPv6 unicast routing.		X
log host ipv6	Configures the IPv6 Syslog server.	X	X
no ipv6 enable	Disables IPv6 on a global basis on a Layer 2 switch.	X	
ping ipv6	Performs an ICMP for IPv6 echo test.	X	X
show ipv6	Displays some global IPv6 parameters, such IPv6 DNS server address.	X	X

Table 28.2: IPv6 CLI Command Support

IPv6 Command	Description	Switch Code	Router Code
show ipv6 access-list	Displays configured IPv6 access lists.	X	X
show ipv6 cache	Displays the IPv6 host cache.		X
show ipv6 interface	Displays IPv6 information for an interface.		X
show ipv6 mld-snooping	Displays information about MLD snooping.	X	X
show ipv6 nd-ns-multicast-macs	Displays IPv6 NS multicast MACs.		X
show ipv6 neighbor	Displays the IPv6 neighbor table.	X	X
show ipv6 ospf	Displays information about OSPF V3.		X
show ipv6 prefix-lists	Displays the configured IPv6 prefix lists.		X
show ipv6 rip	Displays information about RIPng.		X
show ipv6 route	Displays IPv6 routes.		X
show ipv6 router	Displays IPv6 local routers.		X
show ipv6 tcp	Displays information about IPv6 TCP sessions.	X	X
show ipv6 traffic	Displays IPv6 packet counters.	X	X
show ipv6 tunnel	Displays information about IPv6 tunnels	X	X
snmp-client ipv6	Restricts SNMP access to a certain IPv6 node.	X	X
snmp-server host ipv6	Specifies the recipient of SNMP notifications.	X	X
sntp server ipv6	Enables the Foundry device to send SNTP packets over IPv6.	X	X
telnet	Enables a Telnet connection from the Foundry device to a remote IPv6 host using the console.	X	X
traceroute ipv6	Traces a path from the Foundry device to an IPv6 host.	X	X
web access-group ipv6	Restricts Web management access to certain IPv6 hosts as determined by IPv6 ACLs.	X	X
web client ipv6	Restricts Web management access to certain IPv6 hosts.	X	X

Configuring an IPv6 Host Address on a Layer 2 Switch

NOTE: This feature is available on the FastIron X Series only when it is configured as a switch.

In a Layer 3 (router) configuration, each port can be configured separately with an IPv6 address. This is accomplished using the interface configuration process that is described in "Configuring IPv6 on Each Router Interface" on page 28-9.

In a Layer 2 (switch) configuration, individual ports cannot be configured with an IP address (IPv4 or IPv6). In this situation, the switch has one IP address for the management port and one IP address for the system. This has previously been supported for IPv4 but not for IPv6.

There is support for configuring an IPv6 address on the management port as described in "Configuring the Management Port for an IPv6 Automatic Address Configuration" on page 28-8, and for configuring a system-wide IPv6 address on a Layer 2 switch. Configuration of the system-wide IPv6 address is exactly like configuration of an IPv6 address in router mode, except that the IPv6 configuration is at the Global Config level instead of at the Interface Config level.

The process for defining the system-wide interface for IPv6 is described in the following sections:

- "Configuring a Global or Site-Local IPv6 Address with a Manually Configured Interface ID as the Switch's System-wide Address" on page 28-7
- "Configuring a Global or Site-Local IPv6 Address with an Automatically Computed EUI-64 Interface ID as the Switch's System-wide Address" on page 28-7
- "Configuring a Link-Local IPv6 Address as the Switch's System-Wide Address" on page 28-8

Configuring a Global or Site-Local IPv6 Address with a Manually Configured Interface ID as the Switch's System-wide Address

To configure a global or site-local IPv6 Address with a manually configured interface ID, as a switch's system-wide address, enter a command such as the following at the Global Config level:

```
FastIron(config)#ipv6 address 2001:200:12D:1300:240:D0FF:FE48:4000:1/64
```

Syntax: ipv6 address <ipv6-prefix>/<prefix-length>

You must specify the <ipv6-prefix> parameter in hexadecimal using 16-bit values between colons as documented in RFC 2373.

You must specify the <prefix-length> parameter in decimal value. A slash mark (/) must follow the <ipv6-prefix> parameter and precede the <prefix-length> parameter.

Configuring a Global or Site-Local IPv6 Address with an Automatically Computed EUI-64 Interface ID as the Switch's System-wide Address

To configure a global or site-local IPv6 address with an automatically computed EUI-64 interface ID in the low order 64-bits s the system-wide address, enter commands such as the following:

```
FastIron(config)# ipv6 address 2001:200:12D:1300::/64 eui-64
```

These commands configure the global prefix 2001:200:12d:1300::/64 and an interface ID as the system-wide address, and enable IPv6.

Syntax: ipv6 address <ipv6-prefix>/<prefix-length> eui-64

You must specify the <ipv6-prefix> parameter in hexadecimal using 16-bit values between colons as documented in RFC 2373.

You must specify the <prefix-length> parameter as a decimal value. A slash mark (/) must follow the <ipv6-prefix> parameter and precede the <prefix-length> parameter.

The **eui-64** keyword configures the global or site-local address with an EUI-64 interface ID in the low-order 64 bits. The interface ID is automatically constructed in IEEE EUI-64 format using the interface's MAC address.

Configuring a Link-Local IPv6 Address as the Switch's System-Wide Address

To enable IPv6 and automatically configure a global interface enter commands such as the following:

```
FastIron(config)# ipv6 enable
```

This command enables IPv6 on the switch and specifies that the interface is assigned an automatically computed link-local address.

Syntax: [no] ipv6 enable

To override a link-local address that is automatically computed for the global interface with a manually configured address, enter a command such as the following:

```
FastIron(config)# ipv6 address FE80::240:D0FF:FE48:4672 link-local
```

This command explicitly configures the link-local address FE80::240:D0FF:FE48:4672 for the global interface.

Syntax: ipv6 address <ipv6-address> link-local

You must specify the <ipv6-address> parameter in hexadecimal using 16-bit values between colons as documented in RFC 2373.

The **link-local** keyword indicates that the router interface should use the manually configured link-local address instead of the automatically computed link-local address.

Configuring the Management Port for an IPv6 Automatic Address Configuration

You can have the management port configured to automatically obtain an IPv6 address. This process is the same for any other port and is described in detail in the "Configuring a Global or Site-Local IPv6 Address with an Automatically Computed EUI-64 Interface ID" on page 28-10

Configuring Basic IPv6 Connectivity on a Layer 3 Switch

To configure basic IPv6 connectivity on a Foundry Layer 3 Switch, you must do the following:

- Enable IPv6 routing globally on the switch
- Configure an IPv6 address or explicitly enable IPv6 on each router interface over which you plan to forward IPv6 traffic
- Configure IPv4 and IPv6 protocol stacks. (This step is mandatory only if you want a router interface to send and receive both IPv4 and IPv6 traffic.)

All other configuration tasks in this chapter are optional.

Enabling IPv6 Routing

By default, IPv6 routing is disabled. To enable the forwarding of IPv6 traffic globally on the Layer 3 switch, enter the following command:

```
FastIron(config)# ipv6 unicast-routing
```

Syntax: [no] ipv6 unicast-routing

To disable the forwarding of IPv6 traffic globally on the Foundry device, enter the **no** form of this command.

Configuring IPv6 on Each Router Interface

To forward IPv6 traffic on a router interface, the interface must have an IPv6 address, or IPv6 must be explicitly enabled. By default, an IPv6 address is not configured on a router interface.

If you choose to configure a global or site-local IPv6 address for an interface, IPv6 is also enabled on the interface. Further, when you configure a global or site-local IPv6 address, you must decide on one of the following in the low-order 64 bits:

- A manually configured interface ID.
- An automatically computed EUI-64 interface ID.

If you prefer to assign a link-local IPv6 address to the interface, you must explicitly enable IPv6 on the interface, which causes a link-local address to be automatically computed for the interface. If preferred, you can override the automatically configured link-local address with an address that you manually configure.

This section provides the following information:

- Configuring a global or site-local address with a manually configured or automatically computed interface ID for an interface.
- Automatically or manually configuring a link-local address for an interface.
- Configuring IPv6 anycast addresses

Configuring a Global or Site-Local IPv6 Address

Configuring a global or site-local IPv6 address on an interface does the following:

- Automatically configures an interface ID (a link-local address), if specified.
- Enables IPv6 on that interface.

Additionally, the configured interface automatically joins the following required multicast groups for that link:

- Solicited-node multicast group FF02:0:0:0:0:1:FF00::/104 for each unicast address assigned to the interface.
- All-nodes link-local multicast group FF02::1
- All-routers link-local multicast group FF02::2

The neighbor discovery feature sends messages to these multicast groups. For more information, see "Configuring IPv6 Neighbor Discovery" on page 28-30.

Configuring a Global or Site-Local IPv6 Address with a Manually Configured Interface ID

To configure a global or site-local IPv6 address, including a manually configured interface ID, for an interface, enter commands such as the following:

```
FastIron(config)# interface ethernet 3/1
FastIron(config-if-e100-3/1)# ipv6 address 2001:200:12D:1300:240:D0FF:
FE48:4672:/64
```

These commands configure the global prefix 2001:200:12d:1300::/64 and the interface ID ::240:D0FF:FE48:4672, and enable IPv6 on Ethernet interface 3/1.

Syntax: ipv6 address <ipv6-prefix>/<prefix-length>

You must specify the <ipv6-prefix> parameter in hexadecimal using 16-bit values between colons as documented in RFC 2373.

You must specify the <prefix-length> parameter as a decimal value. A slash mark (/) must follow the <ipv6-prefix> parameter and precede the <prefix-length> parameter.

Configuring a Global or Site-Local IPv6 Address with an Automatically Computed EUI-64 Interface ID

To configure a global or site-local IPv6 address with an automatically computed EUI-64 interface ID in the low-order 64-bits, enter commands such as the following:

```
FastIron(config)# interface ethernet 3/1
FastIron(config-if-e100-3/1)# ipv6 address 2001:200:12D:1300::/64 eui-64
```

These commands configure the global prefix 2001:200:12d:1300::/64 and an interface ID, and enable IPv6 on Ethernet interface 3/1.

Syntax: ipv6 address <ipv6-prefix>/<prefix-length> eui-64

You must specify the <ipv6-prefix> parameter in hexadecimal using 16-bit values between colons as documented in RFC 2373.

You must specify the <prefix-length> parameter as a decimal value. A slash mark (/) must follow the <ipv6-prefix> parameter and precede the <prefix-length> parameter.

The **eui-64** keyword configures the global or site-local address with an EUI-64 interface ID in the low-order 64 bits. The interface ID is automatically constructed in IEEE EUI-64 format using the interface's MAC address.

Configuring a Link-Local IPv6 Address

To explicitly enable IPv6 on a router interface without configuring a global or site-local address for the interface, enter commands such as the following:

```
FastIron(config)# interface ethernet 3/1
FastIron(config-if-e100-3/1)# ipv6 enable
```

These commands enable IPv6 on Ethernet interface 3/1 and specify that the interface is assigned an automatically computed link-local address.

Syntax: [no] ipv6 enable

NOTE: When configuring VLANs that share a common tagged interface with a Virtual Ethernet (VE) interface, Foundry recommends that you override the automatically computed link-local address with a manually configured unique address for the interface. If the interface uses the automatically computed address, which in the case of VE interfaces is derived from a global MAC address, all VE interfaces will have the same MAC address.

To override a link-local address that is automatically computed for an interface with a manually configured address, enter commands such as the following:

```
FastIron(config)# interface ethernet 3/1
FastIron(config-if-e100-3/1)# ipv6 address FE80::240:D0FF:FE48:4672 link-local
```

These commands explicitly configure the link-local address FE80::240:D0FF:FE48:4672 for Ethernet interface 3/1.

Syntax: ipv6 address <ipv6-address> link-local

You must specify the <ipv6-address> parameter in hexadecimal using 16-bit values between colons as documented in RFC 2373.

The **link-local** keyword indicates that the router interface should use the manually configured link-local address instead of the automatically computed link-local address.

Configuring IPv6 Anycast Addresses

In IPv6, an ***anycast*** address is an address for a set of interfaces belonging to different nodes. Sending a packet to an anycast address results in the delivery of the packet to the closest interface configured with the anycast address.

An anycast address looks similar to a unicast address, because it is allocated from the unicast address space. If you assign an IPv6 unicast address to multiple interfaces, it is an anycast address. On the Foundry device, you configure an interface assigned an anycast address to recognize the address as an anycast address.

For example, the following commands configure an anycast address on interface 2/1:

```
FastIron(config)# int e 2/1
FastIron(config-if-e100-2/1)# ipv6 address 2002::6/64 anycast
```

Syntax: ipv6 address <ipv6-prefix>/<prefix-length> [anycast]

IPv6 anycast addresses are described in detail in RFC 1884. See RFC 2461 for a description of how the IPv6 Neighbor Discovery mechanism handles anycast addresses.

Configuring IPv4 and IPv6 Protocol Stacks

One situation in which you must configure a router to run both IPv4 and IPv6 protocol stacks is if it is deployed as an endpoint for an IPv6 over IPv4 tunnel.

Each router interface that will send and receive both IPv4 and IPv6 traffic must be configured with an IPv4 address and an IPv6 address. (An alternative to configuring a router interface with an IPv6 address is to explicitly enable IPv6 using the **ipv6 enable** command. For more information about using this command, see "Configuring a Link-Local IPv6 Address" on page 28-10.)

To configure a router interface to support both the IPv4 and IPv6 protocol stacks, use commands such as the following:

```
FastIron(config)# ipv6 unicast-routing
FastIron(config)# interface ethernet 3/1
FastIron(config-if-e100-3/1)# ip address 192.168.1.1 255.255.255.0
FastIron(config-if-e100-3/1)# ipv6 address 2001:200:12d:1300::/64 eui-64
```

These commands globally enable IPv6 routing and configure an IPv4 address and an IPv6 address for Ethernet interface 3/1.

Syntax: [no] ipv6 unicast-routing

To disable IPv6 traffic globally on the router, enter the **no** form of this command.

Syntax: ip address <ip-address> <sub-net-mask> [secondary]

You must specify the <ip-address> parameter using 8-bit values in dotted decimal notation.

You can specify the <sub-net-mask> parameter in either dotted decimal notation or as a decimal value preceded by a slash mark (/).

The **secondary** keyword specifies that the configured address is a secondary IPv4 address.

To remove the IPv4 address from the interface, enter the **no** form of this command.

Syntax: ipv6 address <ipv6-prefix>/<prefix-length> [eui-64]

This syntax specifies a global or site-local IPv6 address. For information about configuring a link-local IPv6 address, see "Configuring a Link-Local IPv6 Address" on page 28-10.

You must specify the <ipv6-prefix> parameter in hexadecimal using 16-bit values between colons as documented in RFC 2373.

You must specify the <prefix-length> parameter as a decimal value. A slash mark (/) must follow the <ipv6-prefix> parameter and precede the <prefix-length> parameter.

The **eui-64** keyword configures the global or site-local address with an EUI-64 interface ID in the low-order 64 bits. The interface ID is automatically constructed in IEEE EUI-64 format using the interface's MAC address. If you do not specify the **eui-64** keyword, you must manually configure the 64-bit interface ID as well as the 64-bit network prefix. For more information about manually configuring an interface ID, see "Configuring a Global or Site-Local IPv6 Address" on page 28-9.

IPv6 Host Support

Platform Support: FastIron X Series devices running software release 02.4.00 and later – L2, BL3, L3

You can configure a FastIron X Series switch to serve as an IPv6 host in an IPv6 network. An ***IPv6 host*** has IPv6 addresses on its interfaces, but does not have full IPv6 routing enabled on it.

This section describes the following IPv6 host features:

- "IPv6 Access Lists"
- "Restricting SNMP Access to an IPv6 Node"
- "Specifying an IPv6 SNMP Trap Receiver"
- "SNMP3 over IPv6"
- "Secure Shell, SCP, and IPv6"
- "IPv6 Telnet"
- "IPv6 Traceroute"
- "IPv6 Web Management using HTTP and HTTPS"
- "Restricting Web Management Access"
- "Configuring Name-to-IPv6 Address Resolution using IPv6 DNS Resolver"
- "Defining an IPv6 DNS Entry"
- "Using the IPv6 copy Command"
- "Using the IPv6 ncopy Command"
- "IPv6 Ping"
- "Configuring an IPv6 Syslog Server"
- "Viewing IPv6 SNMP Server Addresses"
- "Disabling Router Advertisement and Solicitation Messages"
- "IPv6 Debug"
- "Disabling IPv6 on a Layer 2 Switch"

The following IPv6 host features are also supported and are documented elsewhere in this chapter or other chapter(s) of this guide:

- "Configuring a Link-Local IPv6 Address as the Switch's System-Wide Address"
- "SNTP over IPv6" on page 4-11

IPv6 Access Lists

You can configure an IPv6 ACL to filter traffic to or from an IPv6 host. To do so, see "Configuring IPv6 Access Control Lists (ACLs)" on page 18-1,

Restricting SNMP Access to an IPv6 Node

You can restrict SNMP access (which includes IronView Network Manager) to the device to the IPv6 host whose IP address you specify. To do so, enter a command such as the following:

```
FastIron(config)# snmp-client ipv6 2001:efff:89::23
```

Syntax: snmp-client ipv6 <ipv6-address>

The <ipv6-address> you specify must be in hexadecimal format using 16-bit values between colons as documented in RFC 2373.

Specifying an IPv6 SNMP Trap Receiver

You can specify an IPv6 host as a trap receiver to ensure that all SNMP traps sent by the device will go to the same SNMP trap receiver or set of receivers, typically one or more host devices on the network. To do so, enter a command such as the following:

```
FastIron(config)# snmp-server host ipv6 2001:efff:89::13
```

Syntax: snmp-server host ipv6 <ipv6-address>

The <ipv6-address> you specify must be in hexadecimal format using 16-bit values between colons as documented in RFC 2373.

SNMP3 over IPv6

Foundry FastIron devices support IPv6 for SNMP version 3. For more information about how to configure SNMP, see the chapter “Securing SNMP Access” on page 47-1.

Secure Shell, SCP, and IPv6

Secure Shell (SSH) is a mechanism that allows secure remote access to management functions on the Foundry device. SSH provides a function similar to Telnet. You can log in to and configure the Foundry device using a publicly or commercially available SSH client program, just as you can with Telnet. However, unlike Telnet, which provides no security, SSH provides a secure, encrypted connection to the Foundry device.

To open an SSH session between an IPv6 host running an SSH client program and the Foundry device, open the SSH client program and specify the IPv6 address of the device. For more information about configuring SSH on the Foundry device, see “Configuring SSHv2 and SCP” on page 41-1.

IPv6 Telnet

Telnet sessions can be established between a Foundry device to a remote IPv6 host, and from a remote IPv6 host to the Foundry device using IPv6 addresses.

The **telnet** command establishes a Telnet connection from a Foundry device to a remote IPv6 host using the console. Up to five ***read-access*** Telnet sessions are supported on the router at one time. ***Write-access*** through Telnet is limited to one session, and only one outgoing Telnet session is supported on the router at one time. To see the number of open Telnet sessions at any time, enter the **show telnet** command.

EXAMPLES:

To establish a Telnet connection to a remote host with the IPv6 address of 3001:2837:3de2:c37::6, enter the following command:

```
FastIron#telnet 3001:2837:3de2:c37::6
```

Syntax: telnet <ipv6-address> [<port-number> | outgoing-interface ethernet <port> | ve <number>]

The <ipv6-address> parameter specifies the address of a remote host. You must specify this address in hexadecimal using 16-bit values between colons as documented in RFC 2373.

The <port-number> parameter specifies the port number on which the Foundry device establishes the Telnet connection. You can specify a value between 1 - 65535. If you do not specify a port number, the Foundry device establishes the Telnet connection on port 23.

If the IPv6 address you specify is a link-local address, you must specify the **outgoing-interface** ethernet <port> | ve <number> parameter. This parameter identifies the interface that must be used to reach the remote host. If you specify an Ethernet interface, you must also specify the port number associated with the interface. If you specify a VE interface, also specify the VE number.

Establishing a Telnet Session From an IPv6 Host

To establish a Telnet session from an IPv6 host to the Foundry device, open your Telnet application and specify the IPv6 address of the Layer 3 Switch.

IPv6 Traceroute

The **traceroute** command allows you to trace a path from the Foundry device to an IPv6 host.

The CLI displays trace route information for each hop as soon as the information is received. Traceroute requests display all responses to a minimum TTL of 1 second and a maximum TTL of 30 seconds. In addition, if there are multiple equal-cost routes to the destination, the Foundry device displays up to three responses.

For example, to trace the path from the Foundry device to a host with an IPv6 address of 3301:23dd:349e:a384::34, enter the following command:

```
FastIron#traceroute ipv6 3301:23dd:349e:a384::34
```

Syntax: traceroute ipv6 <ipv6-address>

The <ipv6-address> parameter specifies the address of a host. You must specify this address in hexadecimal using 16-bit values between colons as documented in RFC 2373.

IPv6 Web Management using HTTP and HTTPS

When you have an IPv6 management station connected to a switch with an IPv6 address applied to the management port, you can manage the switch from a Web browser by entering one of the following in the browser address field:

http://[<ipv6 address>**]**
or
https://[<ipv6 address>]

NOTE: You must enclose the IPv6 address with square brackets [ ] in order for the Web browser to work.

Restricting Web Management Access

You can restrict Web management access to include only management functions on a Foundry device that is acting as an IPv6 host, or restrict access so that the Foundry host can be reached by a specified IPv6 device.

Restricting Web Management Access by Specifying an IPv6 ACL

You can specify an IPv6 ACL that restricts Web management access to management functions on the device that is acting as the IPv6 host. For example:

```
FastIron(config)# access-list 12 deny host 2000:2383:e0bb::2/128 log
FastIron(config)# access-list 12 deny 30ff:3782::ff89/128 log
FastIron(config)# access-list 12 deny 3000:4828::fe19/128 log
FastIron(config)# access-list 12 permit any
FastIron(config)# web access-group ipv6 12
```

Syntax: web access-group ipv6 <ipv6 ACL name>

where <ipv6 ACL name> is a valid IPv6 ACL.

Restricting Web Management Access to an IPv6 Host

You can restrict Web management access to the device to the IPv6 host whose IP address you specify. No other device except the one with the specified IPv6 address can access the Foundry device’s Web management interface. For example:

```
FastIron(config)# web client ipv6 3000:2383:e0bb::2/128
```

Syntax: web client ipv6 <ipv6-address>

The <ipv6-address> you specify must be in hexadecimal format using 16-bit values between colons as documented in RFC 2373.

Configuring Name-to-IPv6 Address Resolution using IPv6 DNS Resolver

The Domain Name Server (DNS) resolver feature lets you use a host name to perform Telnet, ping, and traceroute commands. You can also define a DNS domain on a Foundry device and thereby recognize all hosts within that domain. After you define a domain name, the Foundry device automatically appends the appropriate domain to the host and forwards it to the domain name server.

For example, if the domain “newyork.com” is defined on a Foundry device, and you want to initiate a ping to host “NYC01” on that domain, you need to reference only the host name in the command instead of the host name and its domain name. For example, you could enter either of the following commands to initiate the ping:

```
FastIron# ping ipv6 nyc01
FastIron# ping ipv6 nyc01.newyork.com
```

Defining an IPv6 DNS Entry

IPv6 defines new DNS record types to resolve queries for domain names to IPv6 addresses, as well as IPv6 addresses to domain names. Foundry devices running IPv6 software support AAAA DNS records, which are defined in RFC 1886.

AAAA DNS records are analogous to the A DNS records used with IPv4. They store a complete IPv6 address in each record. AAAA records have a type value of 28.

To establish an IPv6 DNS entry for the device, enter the following command:

```
FastIron(config)# ipv6 dns domain-name companynet.com
```

Syntax: [no] ipv6 dns domain-name <domain name>

To define an IPv6 DNS server address, enter the following command:

```
FastIron(config)# ipv6 dns server-address 200::1
```

Syntax: [no] ipv6 dns server-address <ipv6-addr> [<ipv6-addr>] [<ipv6-addr>] [<ipv6-addr>]

As an example, in a configuration where ftp6.companynet.com is a server with an IPv6 protocol stack, when a user pings ftp6.companynet.com, the Foundry device attempts to resolve the AAAA DNS record. In addition, if the DNS server does not have an IPv6 address, as long as it is able to resolve AAAA records, it can still respond to DNS queries.

Using the IPv6 copy Command

The **copy** command for IPv6 allows you to do the following:

- Copy a file from a specified source to an IPv6 TFTP server.
- Copy a file from an IPv6 TFTP server to a specified destination.

Copying a File to an IPv6 TFTP Server

You can copy a file from the following sources to an IPv6 TFTP server:

- Flash memory.
- Running configuration.
- Startup configuration.

Copying a File from Flash Memory

For example, to copy the primary or secondary boot image from the device’s flash memory to an IPv6 TFTP server, enter a command such as the following:

```
FastIron# copy flash tftp 2001:7382:e0ff:7837::3 test.img secondary
```

This command copies the secondary boot image named test.img from flash memory to a TFTP server with the IPv6 address of 2001:7382:e0ff:7837::3.

Syntax: copy flash tftp <ipv6-address> <source-file-name> primary | secondary

The <ipv6-address> parameter specifies the address of the TFTP server. You must specify this address in hexadecimal using 16-bit values between colons as documented in RFC 2373.

The <source-file-name> parameter specifies the name of the file you want to copy to the IPv6 TFTP server.

The **primary** keyword specifies the primary boot image, while the **secondary** keyword specifies the secondary boot image.

Copying a File from the Running or Startup Configuration

For example, to copy the running configuration to an IPv6 TFTP server, enter a command such as the following:

```
FastIron# copy running-config tftp 2001:7382:e0ff:7837::3 newrun.cfg
```

This command copies the running configuration to a TFTP server with the IPv6 address of 2001:7382:e0ff:7837::3 and names the file on the TFTP server newrun.cfg.

Syntax: copy running-config | startup-config tftp <ipv6-address> <destination-file-name>

Specify the **running-config** keyword to copy the running configuration file to the specified IPv6 TFTP server.

Specify the **startup-config** keyword to copy the startup configuration file to the specified IPv6 TFTP server.

The tftp <ipv6-address> parameter specifies the address of the TFTP server. You must specify this address in hexadecimal using 16-bit values between colons as documented in RFC 2373.

The <destination-file-name> parameter specifies the name of the file that is copied to the IPv6 TFTP server.

Copying a File from an IPv6 TFTP Server

You can copy a file from an IPv6 TFTP server to the following destinations:

- Flash memory.
- Running configuration.
- Startup configuration.

Copying a File to Flash Memory

For example, to copy a boot image from an IPv6 TFTP server to the primary or secondary storage location in the device's flash memory, enter a command such as the following:

```
FastIron# copy tftp flash 2001:7382:e0ff:7837::3 test.img secondary
```

This command copies a boot image named test.img from an IPv6 TFTP server with the IPv6 address of 2001:7382:e0ff:7837::3 to the secondary storage location in the device's flash memory.

Syntax: copy tftp flash <ipv6-address> <source-file-name> primary | secondary

The <ipv6-address> parameter specifies the address of the TFTP server. You must specify this address in hexadecimal using 16-bit values between colons as documented in RFC 2373.

The <source-file-name> parameter specifies the name of the file you want to copy from the IPv6 TFTP server.

The **primary** keyword specifies the primary storage location in the device's flash memory, while the **secondary** keyword specifies the secondary storage location in the device's flash memory.

Copying a File to the Running or Startup Configuration

For example, to copy a configuration file from an IPv6 TFTP server to the router's running or startup configuration, enter a command such as the following.

```
FastIron# copy tftp running-config 2001:7382:e0ff:7837::3 newrun.cfg overwrite
```

This command copies the newrun.cfg file from the IPv6 TFTP server and overwrites the router's running configuration file with the contents of newrun.cfg.

NOTE: To activate this configuration, you must reload (reset) the device.

Syntax: copy tftp running-config | startup-config <ipv6-address> <source-file-name> [overwrite]

Specify the **running-config** keyword to copy the running configuration from the specified IPv6 TFTP server.

Specify the **startup-config** keyword to copy the startup configuration from the specified IPv6 TFTP server.

The <ipv6-address> parameter specifies the address of the TFTP server. You must specify this address in hexadecimal using 16-bit values between colons as documented in RFC 2373.

The <source-file-name> parameter specifies the name of the file that is copied from the IPv6 TFTP server.

The **overwrite** keyword specifies that the device should overwrite the current configuration file with the copied file. If you do not specify this parameter, the device copies the file into the current running or startup configuration but does not overwrite the current configuration.

NOTE: You cannot use the overwrite option from non-console sessions, because it will disconnect the session.

Using the IPv6 ncopy Command

The **ncopy** command for IPv6 allows you to do the following:

- Copy a primary or secondary boot image from flash memory to an IPv6 TFTP server.
- Copy the running configuration to an IPv6 TFTP server.
- Copy the startup configuration to an IPv6 TFTP server
- Upload various files from an IPv6 TFTP server.

Copying a Primary or Secondary Boot Image from Flash Memory to an IPv6 TFTP Server

For example, to copy the primary or secondary boot image from the device's flash memory to an IPv6 TFTP server, enter a command such as the following:

```
FastIron# ncopy flash primary tftp 2001:7382:e0ff:7837::3 primary.img
```

This command copies the primary boot image named primary.img from flash memory to a TFTP server with the IPv6 address of 2001:7382:e0ff:7837::3.

Syntax: ncopy flash primary | secondary tftp <ipv6-address> <source-file-name>

The **primary** keyword specifies the primary boot image, while the **secondary** keyword specifies the secondary boot image.

The **tftp** <ipv6-address> parameter specifies the address of the TFTP server. You must specify this address in hexadecimal using 16-bit values between colons as documented in RFC 2373.

The <source-file-name> parameter specifies the name of the file you want to copy from flash memory.

Copying the Running or Startup Configuration to an IPv6 TFTP Server

For example, to copy a device's running or startup configuration to an IPv6 TFTP server, enter a command such as the following:

```
FastIron# ncopy running-config tftp 2001:7382:e0ff:7837::3 bakrun.cfg
```

This command copies a device's running configuration to a TFTP server with the IPv6 address of 2001:7382:e0ff:7837::3 and names the destination file bakrun.cfg.

Syntax: ncopy running-config | startup-config tftp <ipv6-address> <destination-file-name>

Specify the **running-config** keyword to copy the device's running configuration or the **startup-config** keyword to copy the device's startup configuration.

The **tftp** <ipv6-address> parameter specifies the address of the TFTP server. You must specify this address in hexadecimal using 16-bit values between colons as documented in RFC 2373.

The <destination-file-name> parameter specifies the name of the running configuration that is copied to the IPv6 TFTP server.

Uploading Files from an IPv6 TFTP Server

You can upload the following files from an IPv6 TFTP server:

- Primary boot image.
- Secondary boot image.
- Running configuration.
- Startup configuration.

Uploading a Primary or Secondary Boot Image from an IPv6 TFTP Server

For example, to upload a primary or secondary boot image from an IPv6 TFTP server to a device's flash memory, enter a command such as the following:

```
FastIron# ncopy tftp 2001:7382:e0ff:7837::3 primary.img flash primary
```

This command uploads the primary boot image named primary.img from a TFTP server with the IPv6 address of 2001:7382:e0ff:7837::3 to the device's primary storage location in flash memory.

Syntax: ncopy tftp <ipv6-address> <source-file-name> flash primary | secondary

The **tftp** <ipv6-address> parameter specifies the address of the TFTP server. You must specify this address in hexadecimal using 16-bit values between colons as documented in RFC 2373.

The <source-file-name> parameter specifies the name of the file you want to copy from the TFTP server.

The **primary** keyword specifies the primary location in flash memory, while the **secondary** keyword specifies the secondary location in flash memory.

Uploading a Running or Startup Configuration from an IPv6 TFTP Server

For example to upload a running or startup configuration from an IPv6 TFTP server to a device, enter a command such as the following:

```
FastIron# ncopy tftp 2001:7382:e0ff:7837::3 newrun.cfg running-config
```

This command uploads a file named newrun.cfg from a TFTP server with the IPv6 address of 2001:7382:e0ff:7837::3 to the device.

Syntax: ncopy tftp <ipv6-address> <source-file-name> running-config | startup-config

The **tftp** <ipv6-address> parameter specifies the address of the TFTP server. You must specify this address in hexadecimal using 16-bit values between colons as documented in RFC 2373.

The <source-file-name> parameter specifies the name of the file you want to copy from the TFTP server.

Specify the **running-config** keyword to upload the specified file from the IPv6 TFTP server to the device. The device copies the specified file into the current running configuration but does not overwrite the current configuration.

Specify the **startup-config** keyword to upload the specified file from the IPv6 TFTP server to the device. The the device copies the specified file into the current startup configuration but does not overwrite the current configuration.

IPv6 Ping

The **ping** command allows you to verify the connectivity from a Foundry device to an IPv6 device by performing an ICMP for IPv6 echo test.

For example, to ping a device with the IPv6 address of 2001:3424:847f:a385:34dd::45 from the Foundry device, enter the following command:

```
FastIron#ping ipv6 2001:3424:847f:a385:34dd::45
```

Syntax: ping ipv6 <ipv6-address> [outgoing-interface [<port> | ve <number>]] [source <ipv6-address>] [count <number>] [timeout <milliseconds>] [ttl <number>] [size <bytes>] [quiet] [numeric] [no-fragment] [verify] [data <1-to-4 byte hex>] [brief]

- The <ipv6-address> parameter specifies the address of the router. You must specify this address in hexadecimal using 16-bit values between colons as documented in RFC 2373.
- The **outgoing-interface** keyword specifies a physical interface over which you can verify connectivity. If you specify a physical interface, such as an Ethernet interface, you must also specify the port number of the interface. If you specify a virtual interface, such as a VE, you must specify the number associated with the VE.
- The **source** <ipv6-address> parameter specifies an IPv6 address to be used as the origin of the ping packets.
- The **count** <number> parameter specifies how many ping packets the router sends. You can specify from 1 - 4294967296. The default is 1.
- The **timeout** <milliseconds> parameter specifies how many milliseconds the router waits for a reply from the pinged device. You can specify a timeout from 1 - 4294967296 milliseconds. The default is 5000 (5 seconds).
- The **ttl** <number> parameter specifies the maximum number of hops. You can specify a TTL from 1 - 255. The default is 64.
- The **size** <bytes> parameter specifies the size of the ICMP data portion of the packet. This is the payload and does not include the header. You can specify from 0 - 4000. The default is 16.
- The **no-fragment** keyword turns on the "don't fragment" bit in the IPv6 header of the ping packet. This option is disabled by default.
- The **quiet** keyword hides informational messages such as a summary of the ping parameters sent to the device, and instead only displays messages indicating the success or failure of the ping. This option is disabled by default.
- The **verify** keyword verifies that the data in the echo packet (the reply packet) is the same as the data in the echo request (the ping). By default the device does not verify the data.
- The **data** <1 - 4 byte hex> parameter lets you specify a specific data pattern for the payload instead of the default data pattern, "abcd", in the packet's data payload. The pattern repeats itself throughout the ICMP message (payload) portion of the packet.

NOTE: For parameters that require a numeric value, the CLI does not check that the value you enter is within the allowed range. Instead, if you do exceed the range for a numeric value, the software rounds the value to the nearest valid value.

- The **brief** keyword causes ping test characters to be displayed. The following ping test characters are supported:

  ! Indicates that a reply was received.

  . Indicates that the network server timed out while waiting for a reply.

  U Indicates that a destination unreachable error PDU was received.

  I Indicates that the user interrupted ping.

Configuring an IPv6 Syslog Server

To enable IPv6 logging, specify an IPv6 Syslog server. Enter a command such as the following:

```
FastIron(config)# log host ipv6 2000:2383:e0bb::4/128
```

Syntax: log host ipv6 <ipv6-address> [<udp-port-num>]

The <ipv6-address> must be in hexadecimal using 16-bit values between colons as documented in RFC 2373.

The <udp-port-num> optional parameter specifies the UDP application port used for the Syslog facility.

Viewing IPv6 SNMP Server Addresses

Some of the **show** commands display IPv6 addresses for IPv6 SNMP servers. The following shows an example output for the **show snmp server** command.

```
FastIron# show snmp server

      Contact:
     Location:
Community(ro): .....


Traps
              Warm/Cold start: Enable
                      Link up: Enable
                    Link down: Enable
               Authentication: Enable
     Locked address violation: Enable
         Power supply failure: Enable
                  Fan failure: Enable
          Temperature warning: Enable
                 STP new root: Enable
          STP topology change: Enable
                         vsrp: Enable

 Total Trap-Receiver Entries: 4

Trap-Receiver IP-Address                Port-Number Community

     1        192.147.201.100               162     .....

     2        4000::200                     162     .....

     3        192.147.202.100               162     .....

     4        3000::200                     162     .....
```

Disabling Router Advertisement and Solicitation Messages

Router advertisement and solicitation messages enable a node on a link to discover the routers on the same link. By default, router advertisement and solicitation messages are permitted on the device. To disable these messages, configure an IPv6 access list that denies them. The following shows an example configuration.

EXAMPLES:

```
FastIron(config)# ipv6 access-list rtradvert
FastIron(config)# deny icmp any any router-advertisement
FastIron(config)# deny icmp any any router-solicitation
FastIron(config)# permit ipv6 any any
```

IPv6 Debug

The **debug ipv6** commands enable the collection of information about IPv6 configurations for troubleshooting.

Syntax: debug ipv6 <address> <cache> <icmp> <mld> <nd> <packet> <ra>

- address - IPv6 address
- cache - IPv6 cache entry

- icmp - ICMPv6
- mld - MLD protocol activity
    - <add-del-oif>[<all><clear>] <clear> <detail> <down-port> <error> <group> <level> <mcache-group> <mcache-source> <packet> <phy-port> <prime-port> <show> <source> <timer> <vlan>
- nd - neighbor discovery
- packet - IPv6 packet
- ra - router add

Disabling IPv6 on a Layer 2 Switch

IPv6 is enabled by default in the Layer 2 switch code. If desired, you can disable IPv6 on a global basis on a device running the switch code. To do so, enter the following command at the Global CONFIG level of the CLI:

```
FastIron(config)# no ipv6 enable
```

Syntax: no ipv6 enable

To re-enable IPv6 after it has been disabled, enter **ipv6 enable**.

NOTE: IPv6 is disabled by default in the router code and must be configured on each interface that will support IPv6.

Configuring a Static IPv6 Route

You can configure a static IPv6 route to be redistributed into a routing protocol, but you cannot redistribute routes learned by a routing protocol into the static IPv6 routing table.

Before configuring a static IPv6 route, you must enable the forwarding of IPv6 traffic on the Layer 3 switch using the **ipv6 unicast-routing** command and enable IPv6 on at least one interface by configuring an IPv6 address or explicitly enabling IPv6 on that interface. For more information on performing these configuration tasks, see “Configuring IPv4 and IPv6 Protocol Stacks” on page 28-11.

To configure a static IPv6 route for a destination network with the prefix 8eff::0/32, a next-hop gateway with the global address 4fee:2343:0:ee44::1, and an administrative distance of 110, enter the following command:

```
FastIron(config)# ipv6 route 8eff::0/32 4fee:2343:0:ee44::1 distance 110
```

Syntax: ipv6 route <dest-ipv6-prefix>/<prefix-length> <next-hop-ipv6-address> [<metric>] [distance <number>]

To configure a static IPv6 route for a destination network with the prefix 8eff::0/32 and a next-hop gateway with the link-local address fe80::1 that the Layer 3 switch can access through Ethernet interface 3/1, enter the following command:

```
FastIron(config)# ipv6 route 8eff::0/32 ethernet 1 fe80::1
```

Syntax: ipv6 route <dest-ipv6-prefix>/<prefix-length> [ethernet <slot/port> | ve <num> | null0] <next-hop-ipv6-address> [<metric>] [distance <number>]

To configure a static IPv6 route for a destination network with the prefix 8eff::0/32 and a next-hop gateway that the Layer 3 switch can access through tunnel 1, enter the following command:

```
FastIron(config)# ipv6 route 8eff::0/32 tunnel 1
```

Syntax: ipv6 route <dest-ipv6-prefix>/<prefix-length> <interface> <port> [<metric>] [distance <number>]

Table 28.3 describes the parameters associated with this command and indicates the status of each parameter.

Table 28.3: Static IPv6 route parameters

Parameter	Configuration Details	Status
The IPv6 prefix and prefix length of the route's destination network.	You must specify the <dest-ipv6-prefix> parameter in hexadecimal using 16-bit values between colons as documented in RFC 2373. You must specify the <prefix-length> parameter as a decimal value. A slash mark (/) must follow the <ipv6-prefix> parameter and precede the <prefix-length> parameter.	Mandatory for all static IPv6 routes.
The route's next-hop gateway, which can be one of the following: • The IPv6 address of a next-hop gateway. • A tunnel interface.	You can specify the next-hop gateway as one of the following types of IPv6 addresses: • A global address. • A link-local address. If you specify a global address, you do not need to specify any additional parameters for the next-hop gateway. If you specify a link-local address, you must also specify the interface through which to access the address. You can specify one of the following interfaces: • An Ethernet interface. • A tunnel interface. • A virtual interface (VE). If you specify an Ethernet interface, also specify the port number associated with the interface. If you specify a VE or tunnel interface, also specify the VE or tunnel number. You can also specify the next-hop gateway as a tunnel interface. If you specify a tunnel interface, also specify the tunnel number.	Mandatory for all static IPv6 routes.
The route's metric.	You can specify a value from 1 – 16.	Optional for all static IPv6 routes. (The default metric is 1.)
The route's administrative distance.	You must specify the **distance** keyword and any numerical value.	Optional for all static IPv6 routes. (The default administrative distance is 1.)

A metric is a value that the Layer 3 switch uses when comparing this route to other static routes in the IPv6 static route table that have the same destination. The metric applies only to routes that the Layer 3 switch has already placed in the IPv6 static route table.

The administrative distance is a value that the Layer 3 switch uses to compare this route with routes from other route sources that have the same destination. (The Layer 3 switch performs this comparison before placing a route in the IPv6 route table.) This parameter does not apply to routes that are already in the IPv6 route table. In general, a low administrative distance indicates a preferred route. By default, static routes take precedence over routes learned by routing protocols. If you want a dynamic route to be chosen over a static route, you can configure the static route with a higher administrative distance than the dynamic route.

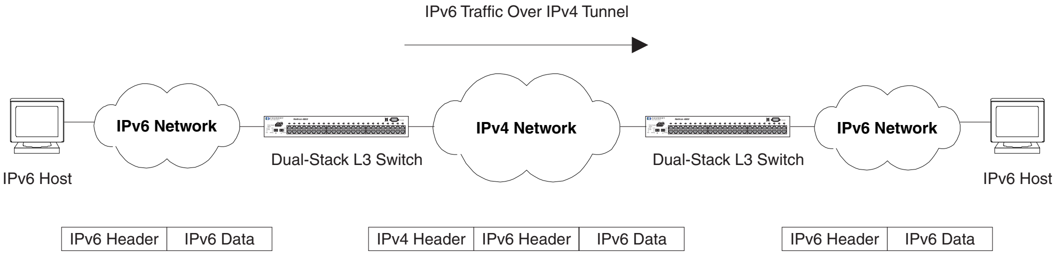
IPv6 Over IPv4 Tunnels in Hardware

Platform Support: FastIron X Series devices running software release 04.1.00 and later – L2, BL3, L3

To enable communication between isolated IPv6 domains using the IPv4 infrastructure, you can manually configure IPv6 over IPv4 tunnels that provide static point-point connectivity.

As shown in Figure 28.2, these tunnels encapsulate an IPv6 packet within an IPv4 packet.

Figure 28.2 IPv6 over an IPv4 Tunnel



In general, a manually configured tunnel establishes a permanent link between switches in IPv6 domains. A manually configured tunnel has explicitly configured IPv4 addresses for the tunnel source and destination.

This tunneling mechanism requires that the Layer 3 switch at each end of the tunnel run both IPv4 and IPv6 protocol stacks. The Layer 3 switches running both protocol stacks, or dual-stack routers, can interoperate directly with both IPv4 and IPv6 end systems and routers. See “Configuring IPv4 and IPv6 Protocol Stacks” on page 28-11.

Configuration Notes

- The local tunnel configuration must include both source and destination addresses.
- The remote side of the tunnel must have the opposite source/destination pair.
- A tunnel interface supports static and dynamic IPv6 configuration settings and routing protocols.
- Duplicate Address Detection (DAD) is not currently supported with IPv6 tunnels. Make sure tunnel endpoints do not have duplicate IP addresses.

Configuring a Manual IPv6 Tunnel

You can use a manually configured tunnel to connect two isolated IPv6 domains. You should deploy this point-to-point tunnelling mechanism if you need a permanent and stable connection.

To configure a manual IPv6 tunnel, enter commands such as the following on a Layer 3 Switch running both IPv4 and IPv6 protocol stacks on each end of the tunnel:

```
FastIron(config)# interface tunnel 1
FastIron(config-tnif-1)#tunnel source ethernet 3/1
FastIron(config-tnif-1)#tunnel destination 198.162.100.1
FastIron(config-tnif-1)#tunnel mode ipv6ip
FastIron(config-tnif-1)#ipv6 enable
```

This example creates tunnel interface 1 and assigns a global IPv6 address with an automatically computed EUI-64 interface ID to it. The IPv4 address assigned to Ethernet interface 3/1 is used as the tunnel source, while the IPv4 address 192.168.100.1 is configured as the tunnel destination. The tunnel mode is specified as a manual

IPv6 tunnel. Finally, the tunnel is enabled. Note that instead of entering **ipv6 enable**, you could specify an IPv6 address, for example, **ipv6 address 2001:b78:384d:34::/64 eui-64**, which would also enable the tunnel.

Syntax: [no] interface tunnel <number>

For the <number> parameter, specify a value between 1 – 8.

Syntax: [no] tunnel source <ipv4-address> | ethernet <port> | loopback <number> | ve <number>

The tunnel source can be an IP address or an interface.

For <ipv4-address>, use 8-bit values in dotted decimal notation.

The **ethernet | loopback | ve** parameter specifies an interface as the tunnel source. If you specify an Ethernet interface, also specify the port number associated with the interface. If you specify a loopback, VE, or interface, also specify the loopback, VE, or number, respectively.

Syntax: [no] tunnel destination <ipv4-address>

Specify the <ipv4-address> parameter using 8-bit values in dotted decimal notation.

Syntax: [no] tunnel mode ipv6ip

ipv6ip indicates that this is an IPv6 manual tunnel.

Syntax: ipv6 enable

The **ipv6 enable** command enables the tunnel. Alternatively, you could specify an IPv6 address, which would also enable the tunnel.

Syntax: ipv6 address <ipv6-prefix>/<prefix-length> [eui-64]

The **ipv6 address** command enables the tunnel. Alternatively, you could enter **ipv6 enable**, which would also enable the tunnel.

Specify the <ipv6-prefix> parameter in hexadecimal format using 16-bit values between colons as documented in RFC 2373.

Specify the <prefix-length> parameter as a decimal value. A slash mark (/) must follow the <ipv6-prefix> parameter and precede the <prefix-length> parameter. The **eui-64** keyword configures the global or site-local address with an EUI-64 interface ID in the low-order 64 bits. The interface ID is automatically constructed in IEEE EUI-64 format using the interface's MAC address.

Clearing IPv6 Tunnel Statistics

You can clear statistics (reset all fields to zero) for all IPv6 tunnels or for a specific tunnel interface.

For example, to clear statistics for tunnel 1, enter the following command at the Privileged EXEC level or any of the Config levels of the CLI:

```
FastIron# clear ipv6 tunnel 1
```

To clear statistics for all IPv6 tunnels, enter the following command:

```
FastIron# clear ipv6 tunnel
```

Syntax: clear ipv6 tunnel [<number>]

The <number> parameter specifies the tunnel number.

Displaying IPv6 Tunnel Information

Use the commands in this section to display the configuration, status, and counters associated with IPv6 tunnels.

Displaying a Summary of Tunnel Information

To display a summary of tunnel information, enter the following command at any level of the CLI:

```
FastIron# show ipv6 tunnel
IP6 Tunnels
  Tunnel  Mode          Packet Received  Packet Sent
  1       configured    0                0
  2       configured    0                22419
```

Syntax: show ipv6 tunnel

This display shows the following information.

Table 28.4: IPv6 Tunnel Summary Information

This Field...	Displays...
Tunnel	The tunnel interface number.
Mode	The tunnel mode. Possible modes include the following: • configured – Indicates a manually configured tunnel.
Packet Received	The number of packets received by a tunnel interface. Note that this is the number of packets received by the CPU. It does not include the number of packets processed in hardware.
Packet Sent	The number of packets sent by a tunnel interface. Note that this is the number of packets sent by the CPU. It does not include the number of packets processed in hardware.

Displaying Tunnel Interface Information

To display status and configuration information for tunnel interface 1, enter the following command at any level of the CLI:

```
FastIron# show interfaces tunnel 1
Tunnel1 is up, line protocol is up
  Hardware is Tunnel
  Tunnel source ethernet 3/5
  Tunnel destination is not configured
  Tunnel mode ipv6ip
  No port name
  MTU 1500 bytes
```

Syntax: show interfaces tunnel <number>

The <number> parameter indicates the tunnel interface number for which you want to display information.

This display shows the following information.

Table 28.5: IPv6 Tunnel Interface Information

This Field...	Displays...
Tunnel interface status	The status of the tunnel interface can be one of the following: • up – The tunnel mode is set and the tunnel interface is enabled. • down – The tunnel mode is not set. • administratively down – The tunnel interface was disabled with the **disable** command.
Line protocol status	The status of the line protocol can be one of the following: • up – IPv4 connectivity is established. • down – The line protocol is not functioning and is down.
Hardware is tunnel	The interface is a tunnel interface.
Tunnel source	The tunnel source can be one of the following: • An IPv4 address • The IPv4 address associated with an interface/port.
Tunnel destination	The tunnel destination can an IPv4 address.
Tunnel mode	The tunnel mode can be one the following: • ipv6ip – indicates a manually configured tunnel
Port name	The port name configured for the tunnel interface.
MTU	The setting of the IPv6 maximum transmission unit (MTU).

Displaying Interface Level IPv6 Settings

To display Interface level IPv6 settings for tunnel interface 1, enter the following command at any level of the CLI:

```
FastIron#show ipv6 inter tunnel 1
Interface Tunnel 1 is up, line protocol is up
  IPv6 is enabled, link-local address is fe80::3:4:2 [Preferred]
  Global unicast address(es):
    1001::1 [Preferred],  subnet is 1001::/64
    1011::1 [Preferred],  subnet is 1011::/64
  Joined group address(es):
    ff02::1:ff04:2
    ff02::5
    ff02::1:ff00:1
    ff02::2
    ff02::1
  MTU is 1480 bytes
  ICMP redirects are enabled
  No Inbound Access List Set
  No Outbound Access List Set
  OSPF enabled
```

The display command above reflects the following configuration:

```
FastIron#show running-config interface tunnel 1
!
interface tunnel 1
 port-name ManualTunnel1
 tunnel mode ipv6ip
 tunnel source loopback 1
 tunnel destination 2.1.1.1
 ipv6 address fe80::3:4:2 link-local
 ipv6 address 1011::1/64
 ipv6 address 1001::1/64
 ipv6 ospf area 0
```

This display shows the following information.

Table 28.6: Interface Level IPv6 Tunnel Information

This Field...	Displays...
Interface Tunnel status	The status of the tunnel interface can be one of the following: • up – IPv4 connectivity is established. • down – The tunnel mode is not set. • administratively down – The tunnel interface was disabled with the **disable** command.
Line protocol status	The status of the line protocol can be one of the following: • up – IPv6 is enabled via the **ipv6 enable** or **ipv6 address** command. • down – The line protocol is not functioning and is down.

ECMP Load Sharing for IPv6

The IPv6 route table selects the best route to a given destination from among the routes in the tables maintained by the configured routing protocols (BGP4, OSPF, static, and so on). The IPv6 route table can contain more than one path to a given destination. When this occurs, the Foundry device selects the path with the lowest cost for insertion into the routing table. If more than one path with the lowest cost exists, all of these paths are inserted into the routing table, subject to the configured maximum number of load sharing paths (by default 4). The device uses ***Equal-Cost Multi-Path (ECMP) load sharing*** to select a path to a destination.

When the device receives traffic for a destination, and the IPv6 route table contains multiple, equal-cost paths to that destination, the device checks the ***IPv6 forwarding cache*** for a forwarding entry for the destination. The IPv6 forwarding cache provides a fast path for forwarding IPv6 traffic. The IPv6 forwarding cache contains entries that associate a destination host or network with a path (next-hop router).

If the IPv6 forwarding cache contains a forwarding entry for the destination, the Foundry device uses the entry to forward the traffic. If the IPv6 forwarding cache does not contain a forwarding entry for the destination, the software selects a path from among the available equal-cost paths to the destination, then creates an entry in the in the cache based on the calculation. Subsequent traffic for the same destination uses the forwarding entry. Entries remain in the IPv6 forwarding cache for one minute, then are aged out.

If the path selected by the device becomes unavailable, its entry in the IPv6 forwarding cache is removed, a new path is selected from the remaining equal-cost paths to the destination, and an entry is created in the IPv6 forwarding cache using the new path.

Foundry devices support the following ECMP load-sharing methods for IPv6 traffic:

- Network-based – The Foundry device distributes traffic across equal-cost paths based on destination network address. The software selects a path based on a calculation involving the maximum number of load-sharing paths allowed and the actual number of paths to the destination network. This is the default ECMP load-sharing method for IPv6.
- Host-based – The Foundry device uses a simple round-robin mechanism to distribute traffic across the equal-cost paths based on destination host IP address. The device uses this ECMP load-sharing method for IPv6 if you explicitly configure it to do so.

You can manually disable or enable ECMP load sharing for IPv6 and specify the number of equal-cost paths the device can distribute traffic across. In addition, you can display information about the status of ECMP load-sharing on the device, as well as the entries in the IPv6 forwarding cache.

Disabling or Re-Enabling ECMP Load Sharing for IPv6

ECMP load sharing for IPv6 is enabled by default. To disable the feature, enter the following command:

```
FastIron(config)# no ipv6 load-sharing
```

If you want to re-enable the feature after disabling it, you must specify the number of load-sharing paths. The maximum number of paths the device supports is a value from 2 – 8. By entering a command such as the following, iPv6 load-sharing will be re-enabled.

```
FastIron(config)# ipv6 load-sharing 4
```

Syntax: [no] ipv6 load-sharing<num>

The <num> parameter specifies the number of paths and can be from 2 – 8. The default is 4.

Changing the Maximum Number of Load Sharing Paths for IPv6

By default, IPv6 ECMP load sharing allows traffic to be balanced across up to four equal paths. You can change the maximum number of paths the device supports to a value from 2 – 8.

To change the number of ECMP load sharing paths for IPv6, enter a command such as the following:

```
FastIron(config)# ipv6 load-sharing 8
```

Syntax: [no] ipv6 load-sharing [<num>]

The <num> parameter specifies the number of paths and can be from 2 – 8. The default is 4.

DHCP Relay Agent for IPv6

A client locates a DHCP server using a reserved, link-scoped multicast address. For this reason, it is a requirement for direct communication between the client and the server that they be attached by the same link. However, in some situations in which ease of management, economy, and scalability is a concern, it is useful to allow a DHCP client to send a message to a DHCP server by using a DHCP relay agent. A DHCP relay agent, which may reside on the clients link, is used to relay messages between the client and the server. A DHCP relay agent is transparent to the client.

When the relay agent receives a message to be relayed from a client to another relay agent, it creates a new Relay-forward message, puts the original DHCP message to relay forward option, and includes its only address and the address it received is in the same option.

Enabling Support for Network-Based ECMP Load Sharing for IPv6

Network-based ECMP load sharing is supported. If this configuration is selected, traffic is distributed across equal-cost paths based on the destination network address. Routes to each network are stored in CAM and accessed when a path to a network is required. Because multiple hosts are likely to reside on a network, this method uses fewer CAM entries than load sharing by host. When you select network-based ECMP load sharing, you can choose either of the following two CAM modes:

- **Dynamic Mode** – In the dynamic mode, routes are entered into the CAM dynamically using a flow-based scheme. In this mode routes are only added to the CAM as they are required. Once routes are added to the CAM, they are subject to being aged-out when they are not in use. Because this mode conserves CAM, it is useful for situations where CAM resources are stressed or limited.
- **Static Mode** – In the static mode, routes are entered into the CAM whenever they are discovered. Routes aren't aged once routes are added to the CAM and they are subject to being aged-out when they are not in use.

Displaying ECMP Load-Sharing Information for IPv6

To display the status of ECMP load sharing for IPv6, enter the following command:

```
FastIron# show ipv6
Global Settings

  unicast-routing enabled, hop-limit 64
  No Inbound Access List Set
  No Outbound Access List Set
  Prefix-based IPv6 Load-sharing is Enabled, Number of load share paths: 4
```

Syntax: show ipv6

You can display the entries in the IPv6 forwarding cache; for example

:

```
FastIron# show ipv6 cache
Total number of cache entries: 10
    IPv6 Address                        Next Hop                     Port
1   5000:2::2                           LOCAL                        tunnel 2
2   2000:4::106                         LOCAL                        ethe 2
3   2000:4::110                         DIRECT                       ethe 2
4   2002:c0a8:46a::1                    LOCAL                        ethe 2
5   fe80::2e0:52ff:fe99:9737            LOCAL                        ethe 2
6   fe80::ffff:ffff:feff:ffff           LOCAL                        loopback 2
7   fe80::c0a8:46a                      LOCAL                        tunnel 2
8   fe80::c0a8:46a                      LOCAL                        tunnel 6
9   2999::1                             LOCAL                        loopback 2
10  fe80::2e0:52ff:fe99:9700            LOCAL                        ethe 1
```

Syntax: show ipv6 cache [<index-number> | <ipv6-prefix>/<prefix-length> | <ipv6-address> | ethernet <port> | ve <number> | tunnel <number>]

Configuring IPv6 ICMP

As with the Internet Control Message Protocol (ICMP) for IPv4, ICMP for IPv6 provides error and informational messages. Foundry's implementation of the stateless auto configuration, neighbor discovery, and path MTU discovery features use ICMP messages.

This section explains how to configure ICMP redirect messages.

Disabling or Reenabling ICMP Redirect Messages

You can disable or re-enable the sending of ICMP redirect messages by a router. By default, a router can send an ICMP redirect message to a neighboring host to inform it of a better first-hop router on a path to a destination. No further configuration is required to enable the sending of ICMP redirect messages. (For more information about how ICMP redirect messages are implemented for IPv6, see “Configuring IPv6 Neighbor Discovery” on page 28-30.)

For example, to disable the sending of ICMP redirect messages on Ethernet interface 3/1, enter the following commands:

```
FastIron(config)# interface ethernet 3/1
FastIron(config-if-e100-3/1)# no ipv6 redirects
```

Syntax: [no] ipv6 redirects

To re-enable the sending of ICMP redirect messages on Ethernet interface 3/1, enter the following commands:

```
FastIron(config)# interface ethernet 3/1
FastIron(config-if-e100-3/1)# ipv6 redirects
```

Use the **show ipv6 interface** <interface> <port-number> command to verify that the sending of ICMP redirect messages is enabled on a particular interface.

Configuring IPv6 Neighbor Discovery

The neighbor discovery feature for IPv6 uses IPv6 ICMP messages to do the following:

- Determine the link-layer address of a neighbor on the same link.
- Verify that a neighbor is reachable.
- Track neighbor routers.

An IPv6 host is required to listen for and recognize the following addresses that identify itself:

- Link-local address.
- Assigned unicast address.
- Loopback address.
- All-nodes multicast address.
- Solicited-node multicast address.
- Multicast address to all other groups to which it belongs.

You can adjust the following IPv6 neighbor discovery features:

- Neighbor solicitation messages for duplicate address detection.
- Router advertisement messages:
  - Interval between router advertisement messages.
  - Value that indicates a router is advertised as a default router (for use by all nodes on a given link).
  - Prefixes advertised in router advertisement messages.
  - Flags for host stateful autoconfiguration.
- Amount of time during which an IPv6 node considers a remote node reachable (for use by all nodes on a given link).

Neighbor Solicitation and Advertisement Messages

Neighbor solicitation and advertisement messages enable a node to determine the link-layer address of another node (neighbor) on the same link. (This function is similar to the function provided by the Address Resolution Protocol [ARP] in IPv4.) For example, node 1 on a link wants to determine the link-layer address of node 2 on the same link. To do so, node 1, the source node, multicasts a neighbor solicitation message. The neighbor solicitation message, which has a value of 135 in the Type field of the ICMP packet header, contains the following information:

- Source address: IPv6 address of node 1 interface that sends the message.
- Destination address: solicited-node multicast address (FF02:0:0:0:0:1:FF00::/104) that corresponds the IPv6 address of node 2.
- Link-layer address of node 1.
- A query for the link-layer address of node 2.

After receiving the neighbor solicitation message from node 1, node 2 replies by sending a neighbor advertisement message, which has a value of 136 in the Type field of the ICMP packet header. The neighbor solicitation message contains the following information:

- Source address: IPv6 address of the node 2 interface that sends the message.
- Destination address: IPv6 address of node 1.
- Link-layer address of node 2.

After node 1 receives the neighbor advertisement message from node 2, nodes 1 and 2 can now exchange packets on the link.

After the link-layer address of node 2 is determined, node 1 can send neighbor solicitation messages to node 2 to verify that it is reachable. Also, nodes 1, 2, or any other node on the same link can send a neighbor advertisement message to the all-nodes multicast address (FF02::1) if there is a change in their link-layer address.

Router Advertisement and Solicitation Messages

Router advertisement and solicitation messages enable a node on a link to discover the routers on the same link.

Each configured router interface on a link sends out a router advertisement message, which has a value of 134 in the Type field of the ICMP packet header, periodically to the all-nodes link-local multicast address (FF02::1).

A configured router interface can also send a router advertisement message in response to a router solicitation message from a node on the same link. This message is sent to the unicast IPv6 address of the node that sent the router solicitation message.

At system startup, a host on a link sends a router solicitation message to the all-routers multicast address (FF01). Sending a router solicitation message, which has a value of 133 in the Type field of the ICMP packet header, enables the host to automatically configure its IPv6 address immediately instead of awaiting the next periodic router advertisement message.

Because a host at system startup typically does not have a unicast IPv6 address, the source address in the router solicitation message is usually the unspecified IPv6 address (0:0:0:0:0:0:0:0). If the host has a unicast IPv6 address, the source address is the unicast IPv6 address of the host interface sending the router solicitation message.

Entering the **ipv6 unicast-routing** command automatically enables the sending of router advertisement messages on all configured router Ethernet interfaces. You can configure several router advertisement message parameters. For information about disabling the sending of router advertisement messages and the router advertisement parameters that you can configure, see “Enabling and Disabling IPv6 Router Advertisements” on page 28-34 and “Setting IPv6 Router Advertisement Parameters” on page 28-32.

Neighbor Redirect Messages

After forwarding a packet, by default, a router can send a neighbor redirect message to a host to inform it of a better first-hop router. The host receiving the neighbor redirect message will then readdress the packet to the better router.

A router sends a neighbor redirect message only for unicast packets, only to the originating node, and to be processed by the node.

A neighbor redirect message has a value of 137 in the Type field of the ICMP packet header.

Setting Neighbor Solicitation Parameters for Duplicate Address Detection

Although the stateless auto configuration feature assigns the 64-bit interface ID portion of an IPv6 address using the MAC address of the host's NIC, duplicate MAC addresses can occur. Therefore, the duplicate address detection feature verifies that a unicast IPv6 address is unique before it is assigned to a host interface by the stateless auto configuration feature. Duplicate address detection verifies that a unicast IPv6 address is unique.

If duplicate address detection identifies a duplicate unicast IPv6 address, the address is not used. If the duplicate address is the link-local address of the host interface, the interface stops processing IPv6 packets.

You can configure the following neighbor solicitation message parameters that affect duplicate address detection while it verifies that a tentative unicast IPv6 address is unique:

- The number of consecutive neighbor solicitation messages that duplicate address detection sends on an interface. By default, duplicate address detection sends three neighbor solicitation messages without any follow-up messages.
- The interval in seconds at which duplicate address detection sends a neighbor solicitation message on an interface. By default, duplicate address detection sends a neighbor solicitation message every 1 second.

NOTE: For the interval at which duplicate address detection sends a neighbor solicitation message on an interface, the Foundry device uses seconds as the unit of measure instead of milliseconds.

For example, to change the number of neighbor solicitation messages sent on Ethernet interface 3/1 to two and the interval between the transmission of the two messages to 9 seconds, enter the following commands:

```
FastIron(config)# interface ethernet 3/1
FastIron(config-if-e100-3/1)# ipv6 nd dad attempt 2
FastIron(config-if-e100-3/1)# ipv6 nd ns-interval 9
```

Syntax: [no] ipv6 nd dad attempt <number>

Syntax: [no] ipv6 nd ns-interval <number>

For the number of neighbor solicitation messages, you can specify any number of attempts. Configuring a value of 0 disables duplicate address detection processing on the specified interface. To restore the number of messages to the default value, use the **no** form of this command.

For the interval between neighbor solicitation messages, you can specify any number of seconds. Foundry does not recommend very short intervals in normal IPv6 operation. When a non-default value is configured, the configured time is both advertised and used by the router itself. To restore the default interval, use the **no** form of this command.

Setting IPv6 Router Advertisement Parameters

You can adjust the following parameters for router advertisement messages:

- The interval (in seconds) at which an interface sends router advertisement messages. By default, an interface sends a router advertisement message every 200 seconds.
- The "router lifetime" value, which is included in router advertisements sent from a particular interface. The value (in seconds) indicates if the router is advertised as a default router on this interface. If you set the value of this parameter to 0, the router is not advertised as a default router on an interface. If you set this parameter to a value that is not 0, the router is advertised as a default router on this interface. By default, the router lifetime value included in router advertisement messages sent from an interface is 1800 seconds.

When adjusting these parameter settings, Foundry recommends that the interval between router advertisement transmission be less than or equal to the router lifetime value if the router is advertised as a default router. For example, to adjust the interval of router advertisements to 300 seconds and the router lifetime value to 1900 seconds on Ethernet interface 3/1, enter the following commands:

```
FastIron(config)# interface ethernet 3/1
FastIron(config-if-e100-3/1)# ipv6 nd ra-interval 300
FastIron(config-if-e100-3/1)# ipv6 nd ra-lifetime 1900
```

Syntax: [no] ipv6 nd ra-interval <number>

Syntax: [no] ipv6 nd ra-lifetime <number>

The <number> parameter in both commands indicates any numerical value. To restore the default interval or router lifetime value, use the **no** form of the respective command.

Controlling Prefixes Advertised in IPv6 Router Advertisement Messages

By default, router advertisement messages include prefixes configured as addresses on router interfaces using the **ipv6 address** command. You can use the **ipv6 nd prefix-advertisement** command to control exactly which prefixes are included in router advertisement messages. Along with which prefixes the router advertisement messages contain, you can also specify the following parameters:

- Valid lifetime—(Mandatory) The time interval (in seconds) in which the specified prefix is advertised as valid. The default is 2592000 seconds (30 days). When the timer expires, the prefix is no longer considered to be valid.
- Preferred lifetime—(Mandatory) The time interval (in seconds) in which the specified prefix is advertised as preferred. The default is 604800 seconds (7 days). When the timer expires, the prefix is no longer considered to be preferred.
- Onlink flag—(Optional) If this flag is set, the specified prefix is assigned to the link upon which it is advertised. Nodes sending traffic to addresses that contain the specified prefix consider the destination to be reachable on the local link.
- Autoconfiguration flag—(Optional) If this flag is set, the stateless auto configuration feature can use the specified prefix in the automatic configuration of 128-bit IPv6 addresses for hosts on the local link.

For example, to advertise the prefix 2001:e077:a487:7365::/64 in router advertisement messages sent out on Ethernet interface 3/1 with a valid lifetime of 1000 seconds, a preferred lifetime of 800 seconds, and the Onlink and Autoconfig flags set, enter the following commands:

```
FastIron(config)# interface ethernet 3/1
FastIron(config-if-e100-3/1)# ipv6 nd prefix-advertisement 2001:e077:a487:7365::/64
1000 800 onlink autoconfig
```

Syntax: [no] ipv6 nd prefix-advertisement <ipv6-prefix>/<prefix-length> <valid-lifetime> <preferred-lifetime> [autoconfig] [onlink]

You must specify the <ipv6-prefix> parameter in hexadecimal using 16-bit values between colons as documented in RFC 2373.

You must specify the <prefix-length> parameter as a decimal value. A slash mark (/) must follow the <ipv6-prefix> parameter and precede the <prefix-length> parameter.

The valid lifetime and preferred lifetime is a numerical value between 0 – 4294967295 seconds. The default valid lifetime is 2592000 seconds (30 days), while the default preferred lifetime is 604800 seconds (7 days).

To remove a prefix from the router advertisement messages sent from a particular interface, use the **no** form of this command.

Setting Flags in IPv6 Router Advertisement Messages

An IPv6 router advertisement message can include the following flags:

- Managed Address Configuration—This flag indicates to hosts on a local link if they should use the stateful autoconfiguration feature to get IPv6 addresses for their interfaces. If the flag is set, the hosts use stateful autoconfiguration to get addresses as well as non-IPv6-address information. If the flag is not set, the hosts do not use stateful autoconfiguration to get addresses and if the hosts can get non-IPv6-address information from stateful autoconfiguration is determined by the setting of the Other Stateful Configuration flag.
- Other Stateful Configuration—This flag indicates to hosts on a local link if they can get non-IPv6 address autoconfiguration information. If the flag is set, the hosts can use stateful autoconfiguration to get non-IPv6-address information.

NOTE: When determining if hosts can use stateful autoconfiguration to get non-IPv6-address information, a set Managed Address Configuration flag overrides an unset Other Stateful Configuration flag. In this situation, the hosts can obtain nonaddress information. However, if the Managed Address Configuration flag is not set and the Other Stateful Configuration flag is set, then the setting of the Other Stateful Configuration flag is used.

By default, the Managed Address Configuration and Other Stateful Configuration flags are not set in router advertisement messages. For example, to set these flags in router advertisement messages sent from Ethernet interface 3/1, enter the following commands:

```
FastIron(config)# interface ethernet 3/1
FastIron(config-if-e100-3/1)# ipv6 nd managed-config-flag
FastIron(config-if-e100-3/1)# ipv6 nd other-config-flag
```

Syntax: [no] ipv6 nd managed-config-flag

Syntax: [no] ipv6 nd other-config-flag

To remove either flag from router advertisement messages sent on an interface, use the **no** form of the respective command.

Enabling and Disabling IPv6 Router Advertisements

If IPv6 unicast routing is enabled on an Ethernet interface, by default, this interface sends IPv6 router advertisement messages. However, by default, non-LAN interface types, for example, tunnel interfaces, do not send router advertisement messages.

To disable the sending of router advertisement messages on an Ethernet interface, enter commands such as the following:

```
FastIron(config)# interface ethernet 3/1
FastIron(config-if-e100-3/1)# ipv6 nd suppress-ra
```

To enable the sending of router advertisement messages on a tunnel interface, enter commands such as the following:

```
FastIron(config)# interface tunnel 1
FastIron(config-tnif-1)# no ipv6 nd suppress-ra
```

Syntax: [no] ipv6 nd suppress-ra

Configuring Reachable Time for Remote IPv6 Nodes

You can configure the duration (in seconds) that a router considers a remote IPv6 node reachable. By default, a router interface uses the value of 30 seconds.

The router advertisement messages sent by a router interface include the amount of time specified by the **ipv6 nd reachable-time** command so that nodes on a link use the same reachable time duration. By default, the messages include a default value of 0.

NOTE: For the interval at which a router interface sends router advertisement messages, Foundry uses seconds as the unit of measure instead of milliseconds.

Foundry does not recommend configuring a short reachable time duration, because a short duration causes the IPv6 network devices to process the information at a greater frequency.

For example, to configure the reachable time of 40 seconds for Ethernet interface 3/1, enter the following commands:

```
FastIron(config)# interface ethernet 3/1
FastIron(config-if-e100-3/1)# ipv6 nd reachable-time 40
```

Syntax: [no] ipv6 nd reachable-time <seconds>

For the <seconds> parameter, you can specify any numerical value. To restore the default time, use the **no** form of this command.

NOTE: The actual reachable time will be from .5 to 1.5 times the configured or default value.

Changing the IPv6 MTU

The IPv6 MTU is the maximum length of an IPv6 packet that can be transmitted on a particular interface. If an IPv6 packet is longer than an MTU, the host that originated the packet fragments the packet and transmits its contents in multiple packets that are shorter than the configured MTU. You can configure the MTU on individual interfaces. Per RFC 2460, the minimum IPv6 MTU for any interface is 1280 bytes.

For example, to configure the MTU on Ethernet interface 3/1 as 1280 bytes, enter the following commands:

```
FastIron(config)# interface ethernet 3/1
FastIron(config-if-e100-3/1)# ipv6 mtu 1280
```

Syntax: [no] ipv6 mtu <bytes>

You can specify between 1280 – 1500 bytes. If a nondefault value is configured for an interface, router advertisements include an MTU option.

You can configure IPv6 MTU for to be greater than 1500 bytes, although the default remains at 1500 bytes. The value of the MTU you can define depends on the following:

- For a physical port, the maximum value of the MTU is the equal to the maximum frame size of the port minus 18 (Layer 2 MAC header + CRC).
- For a virtual routing interface, the maximum value of the MTU is the maximum frame size configured for the VLAN to which it is associated, minus 18 (Layer 2 MAC header + CRC). If a maximum frame size for a VLAN is not configured, then configure the MTU based on the smallest maximum frame size of all the ports of the VLAN that corresponds to the virtual routing interface, minus 18 (Layer 2 MAC header + CRC).

To define IPv6 MTU globally, enter:

```
FastIron(config)#ipv6 mtu 1300
```

To define IPv6 MTU on an interface, enter:

```
FastIron(config-if-e1000-2/1)#ipv6 mtu
```

Syntax: ipv6 mtu <value>

NOTE: If a the size of a jumbo packet received on a port is equal to the maximum frame size – 18 (Layer 2 MAC header + CRC) and if this value is greater than the outgoing port's IPv4/IPv6 MTU, then it will be forwarded in the CPU.

Configuring Static Neighbor Entries

In some special cases, a neighbor cannot be reached using the neighbor discovery feature. In this situation, you can add a static entry to the IPv6 neighbor discovery cache, which causes a neighbor to be reachable at all times without using neighbor discovery. (A static entry in the IPv6 neighbor discovery cache functions like a static ARP entry in IPv4.)

For example, to add a static entry for a neighbor with the IPv6 address 3001:ffe0:2678:47b and link-layer address 0004.6a2b.8641 that is reachable through Ethernet interface 3/1, enter the following command:

```
FastIron(config)# ipv6 neighbor 3001:ffe0:2678:47b ethernet 3/1 0004.6a2b.8641
```

Syntax: [no] ipv6 neighbor <ipv6-address> ethernet <port> | ve <ve-number> [ethernet <port>] <link-layer-address>

The <ipv6-address> parameter specifies the address of the neighbor.

The **ethernet** | **ve** parameter specifies the interface through which to reach a neighbor. If you specify an Ethernet interface, specify the port number of the Ethernet interface. If you specify a VE, specify the VE number and then the Ethernet port numbers associated with the VE. The link-layer address is a 48-bit hardware address of the neighbor.

If you attempt to add an entry that already exists in the neighbor discovery cache, the software changes the already existing entry to a static entry.

To remove a static IPv6 entry from the IPv6 neighbor discovery cache, use the **no** form of this command.

Limiting the Number of Hops an IPv6 Packet Can Traverse

By default, the maximum number of hops an IPv6 packet can traverse is 64. You can change this value to between 1 – 255 hops. For example, to change the maximum number of hops to 70, you can enter the following command:

```
FastIron(config)# ipv6 hop-limit 70
```

Syntax: [no] ipv6 hop-limit <number>

The number of hops can be from 1 – 255.

Clearing Global IPv6 Information

You can clear the following global IPv6 information:

- Entries from the IPv6 cache.
- Entries from the IPv6 neighbor table.
- IPv6 routes from the IPv6 route table.
- IPv6 traffic statistics.

Clearing the IPv6 Cache

You can remove all entries from the IPv6 cache or specify an entry based on the following:

- IPv6 prefix.
- IPv6 address.
- Interface type.

For example, to remove entries for IPv6 address 2000:e0ff::1, enter the following command at the Privileged EXEC level or any of the Config levels of the CLI:

```
FastIron# clear ipv6 cache 2000:e0ff::1
```

Syntax: clear ipv6 cache [<ipv6-prefix>/<prefix-length> | <ipv6-address> | ethernet <port> | tunnel <number> | ve <number>]

You must specify the <ipv6-prefix> parameter in hexadecimal using 16-bit values between colons as documented in RFC 2373. You must specify the <prefix-length> parameter as a decimal value. A slash mark (/) must follow the <ipv6-prefix> parameter and precede the <prefix-length> parameter.

You must specify the <ipv6-address> parameter in hexadecimal using 16-bit values between colons as documented in RFC 2373.

The **ethernet | tunnel | ve** parameter specifies the interfaces for which you can remove cache entries. If you specify an Ethernet interface, also specify the port number associated with the interface. If you specify a VE or tunnel interface, also specify the VE or tunnel number, respectively.

Clearing IPv6 Neighbor Information

You can remove all entries from the IPv6 neighbor table or specify an entry based on the following:

- IPv6 prefix.

- IPv6 address.
- Interface type.

For example, to remove entries for Ethernet interface 3/1, enter the following command at the Privileged EXEC level or any of the CONFIG levels of the CLI:

```
FastIron# clear ipv6 neighbor ethernet 3/1
```

Syntax: clear ipv6 neighbor [<ipv6-prefix>/<prefix-length> | <ipv6-address> | ethernet <port> | ve <number>]

You must specify the <ipv6-prefix> parameter in hexadecimal using 16-bit values between colons as documented in RFC 2373. You must specify the <prefix-length> parameter as a decimal value. A slash mark (/) must follow the <ipv6-prefix> parameter and precede the <prefix-length> parameter.

You must specify the <ipv6-address> parameter in hexadecimal using 16-bit values between colons as documented in RFC 2373.

The **ethernet | ve** parameter specifies the interfaces for which you can remove cache entries. If you specify an Ethernet interface, also specify the port number associated with the interface. If you specify a VE, also specify the VE number.

Clearing IPv6 Routes from the IPv6 Route Table

You can clear all IPv6 routes or only those routes associated with a particular IPv6 prefix from the IPv6 route table and reset the routes.

For example, to clear IPv6 routes associated with the prefix 2000:7838::/32, enter the following command at the Privileged EXEC level or any of the Config levels of the CLI:

```
FastIron# clear ipv6 route 2000:7838::/32
```

Syntax: clear ipv6 route [<ipv6-prefix>/<prefix-length>]

The <ipv6-prefix>/<prefix-length> parameter clears routes associated with a particular IPv6 prefix. You must specify the <ipv6-prefix> parameter in hexadecimal using 16-bit values between colons as documented in RFC 2373. You must specify the <prefix-length> parameter as a decimal value. A slash mark (/) must follow the <ipv6-prefix> parameter and precede the <prefix-length> parameter.

Clearing IPv6 Traffic Statistics

To clear all IPv6 traffic statistics (reset all fields to zero), enter the following command at the Privileged EXEC level or any of the Config levels of the CLI:

```
FastIron(config)# clear ipv6 traffic
```

Syntax: clear ipv6 traffic

Displaying Global IPv6 Information

You can display output for the following global IPv6 parameters:

- IPv6 cache.
- IPv6 interfaces
- IPv6 neighbors
- IPv6 route table
- Local IPv6 routers
- IPv6 TCP connections and the status of individual connections
- IPv6 traffic statistics

Displaying IPv6 Cache Information

The IPv6 cache contains an IPv6 host table that has indices to the next hop gateway and the router interface on which the route was learned.

To display IPv6 cache information, enter the following command at any CLI level:

```
FastIron# show ipv6 cache
Total number of cache entries: 10
    IPv6 Address                       Next Hop                      Port
1   5000:2::2                          LOCAL                         tunnel 2
2   2000:4::106                        LOCAL                         ethe 3/2
3   2000:4::110                        DIRECT                        ethe 3/2
4   2002:c0a8:46a::1                   LOCAL                         ethe 3/2
5   fe80::2e0:52ff:fe99:9737           LOCAL                         ethe 3/2
6   fe80::ffff:ffff:feff:ffff          LOCAL                         loopback 2
7   fe80::c0a8:46a                     LOCAL                         tunnel 2
8   fe80::c0a8:46a                     LOCAL                         tunnel 6
9   2999::1                            LOCAL                         loopback 2
10  fe80::2e0:52ff:fe99:9700           LOCAL                         ethe 3/1
```

Syntax: show ipv6 cache [<index-number> | <ipv6-prefix>/<prefix-length> | <ipv6-address> | ethernet <port> | ve <number> | tunnel <number>]

The <index-number> parameter restricts the display to the entry for the specified index number and subsequent entries.

The <ipv6-prefix>/<prefix-length> parameter restricts the display to the entries for the specified IPv6 prefix. You must specify the <ipv6-prefix> parameter in hexadecimal using 16-bit values between colons as documented in RFC 2373. You must specify the <prefix-length> parameter as a decimal value. A slash mark (/) must follow the <ipv6-prefix> parameter and precede the <prefix-length> parameter.

The **ethernet | ve | tunnel** parameter restricts the display to the entries for the specified interface. The <ipv6-address> parameter restricts the display to the entries for the specified IPv6 address. You must specify this parameter in hexadecimal using 16-bit values between colons as documented in RFC 2373.

If you specify an Ethernet interface, also specify the port number associated with the interface. If you specify a VE interface, also specify the VE number. If you specify a tunnel interface, also specify the tunnel number.

This display shows the following information:

Table 28.7:IPv6 cache information fields

This Field...	Displays...
Total number of cache entries	The number of entries in the cache table.
IPv6 Address	The host IPv6 address.
Next Hop	The next hop, which can be one of the following: • Direct – The next hop is directly connected to the router. • Local – The next hop is originated on this router. • <ipv6 address> – The IPv6 address of the next hop.
Port	The port on which the entry was learned.

Displaying IPv6 Interface Information

To display IPv6 interface information, enter the following command at any CLI level:

```
FastIron# show ipv6 interface
Routing Protocols : R - RIP  O - OSPF
Interface      Status     Routing  Global Unicast Address
Ethernet 3/3   down/down  R
Ethernet 3/5   down/down
Ethernet 3/17  up/up               2017::c017:101/64
Ethernet 3/19  up/up               2019::c019:101/64
VE 4           down/down
VE 14          up/up               2024::c060:101/64
Loopback 1     up/up               ::1/128
Loopback 2     up/up               2005::303:303/128
Loopback 3     up/up
```

Syntax: show ipv6 interface [<interface> [<port-number> |<number>]]

The <interface> parameter displays detailed information for a specified interface. For the interface, you can specify the **Ethernet**, **loopback**, **tunnel**, or **VE** keywords. If you specify an Ethernet interface, also specify the port number associated with the interface. If you specify a loopback, tunnel, or VE interface, also specify the number associated with the interface.

This display shows the following information:

Table 28.8:General IPv6 interface information fields

This Field...	Displays...
Routing protocols	A one-letter code that represents a routing protocol that can be enabled on an interface.
Interface	The interface type, and the port number or number of the interface.
Status	The status of the interface. The entry in the Status field will be either “up/up” or “down/down”.
Routing	The routing protocols enabled on the interface.
Global Unicast Address	The global unicast address of the interface.

To display detailed information for a specific interface, enter a command such as the following at any CLI level:

```
FastIron# show ipv6 interface ethernet 3/1
Interface Ethernet 3/1 is up, line protocol is up
  IPv6 is enabled, link-local address is fe80::2e0:52ff:fe99:97
  Global unicast address(es):
  Joined group address(es):
    ff02::9
    ff02::1:ff99:9700
    ff02::2
    ff02::1
  MTU is 1500 bytes
  ICMP redirects are enabled
  ND DAD is enabled, number of DAD attempts: 3
  ND reachable time is 30 seconds
  ND advertised reachable time is 0 seconds
  ND retransmit interval is 1 seconds
  ND advertised retransmit interval is 0 seconds
  ND router advertisements are sent every 200 seconds
  ND router advertisements live for 1800 seconds
  No Inbound Access List Set
  No Outbound Access List Set
  RIP enabled
```

This display shows the following information:

Table 28.9:Detailed IPv6 interface information fields

This Field...	Displays...
Interface/line protocol status	The status of interface and line protocol. If you have disabled the interface with the **disable** command, the status will be “administratively down”. Otherwise, the status is either “up” or “down”.
IPv6 status/link-local address	The status of IPv6. The status is either “enabled” or “disabled”. Displays the link-local address, if one is configured for the interface.
Global unicast address(es)	Displays the global unicast address(es), if one or more are configured for the interface.
Joined group address(es)	The multicast address(es) that a router interface listens for and recognizes.
MTU	The setting of the maximum transmission unit (MTU) configured for the IPv6 interface. The MTU is the maximum length an IPv6 packet can have to be transmitted on the interface. If an IPv6 packet is longer than an MTU, the host that originated the packet fragments the packet and transmits its contents in multiple packets that are shorter than the configured MTU.
ICMP	The setting of the ICMP redirect parameter for the interface.
ND	The setting of the various neighbor discovery parameters for the interface.
Access List	The inbound and outbound access lists applied to the interface.
Routing protocols	The routing protocols enabled on the interface.

Displaying IPv6 Neighbor Information

You can display the IPv6 neighbor table, which contains an entry for each IPv6 neighbor with which the router exchanges IPv6 packets.

To display the IPv6 neighbor table, enter the following command at any CLI level:

```
FastIron(config)# show ipv6 neighbor
Total number of Neighbor entries: 3

    IPv6 Address                       LinkLayer-Addr State Age Port    IsR
1   2000:4::110                        00e0.5291.bb37 REACH 20  ethe 3/1  1
2   fe80::2e0:52ff:fe91:bb37           00e0.5291.bb37 DELAY 1   ethe 3/2  1
3   fe80::2e0:52ff:fe91:bb40          00e0.5291.bb40 STALE 5930 ethe 3/3  1
```

Syntax: show ipv6 neighbor [<ipv6-prefix>/<prefix-length> | <ipv6-address> | <interface> [<port> |<number>]]

The <ipv6-prefix>/<prefix-length> parameters restrict the display to the entries for the specified IPv6 prefix. You must specify the <ipv6-prefix> parameter in hexadecimal using 16-bit values between colons as documented in RFC 2373. You must specify the <prefix-length> parameter as a decimal value. A slash mark (/) must follow the <ipv6-prefix> parameter and precede the <prefix-length> parameter.

The <ipv6-address> parameter restricts the display to the entries for the specified IPv6 address. You must specify this parameter in hexadecimal using 16-bit values between colons as documented in RFC 2373.

The <interface> parameter restricts the display to the entries for the specified router interface. For this parameter, you can specify the **Ethernet** or **VE** keywords. If you specify an Ethernet interface, also specify the port number associated with the interface. If you specify a VE interface, also specify the VE number.

This display shows the following information:

l

Table 28.10:IPv6 neighbor information fields

This Field...	Displays...
Total number of neighbor entries	The total number of entries in the IPv6 neighbor table.
IPv6 Address	The 128-bit IPv6 address of the neighbor.
Link-Layer Address	The 48-bit interface ID of the neighbor.
State	The current state of the neighbor. Possible states are as follows: • INCOMPLETE – Address resolution of the entry is being performed. • REACH – The forward path to the neighbor is functioning properly. • STALE – This entry has remained unused for the maximum interval. While stale, no action takes place until a packet is sent. • DELAY – This entry has remained unused for the maximum interval, and a packet was sent before another interval elapsed. • PROBE – Neighbor solicitation are transmitted until a reachability confirmation is received.
Age	The number of seconds the entry has remained unused. If this value remains unused for the number of seconds specified by the **ipv6 nd reachable-time command** (the default is 30 seconds), the entry is removed from the table.
Port	The physical port on which the entry was learned.
IsR	Determines if the neighbor is a router or host: 0 – Indicates that the neighbor is a host. 1 – Indicates that the neighbor is a router.

Displaying the IPv6 Route Table

To display the IPv6 route table, enter the following command at any CLI level:

```
FastIron# show ipv6 route
IPv6 Routing Table - 7 entries:

Type Codes:  C - Connected, S - Static, R - RIP, O - OSPF,

Type IPv6 Prefix                Next Hop Router           Interface  Dis/Metric
C   2000:4::/64                 ::                        ethe 3/2    0/0
S   2002::/16                   ::                        tunnel 6    1/1
S   2002:1234::/32              ::                        tunnel 6    1/1
C   2002:c0a8:46a::/64          ::                        ethe 3/2    0/0
C   2999::1/128                 ::                        loopback 2  0/0
O   2999::2/128                 fe80::2e0:52ff:fe91:bb37  ethe 3/2    110/1
C   5000:2::/64                 ::                        tunnel 2    0/0
```

Syntax: show ipv6 route [<ipv6-address> | <ipv6-prefix>/<prefix-length> | connect | ospf | rip | static | summary]

The <ipv6-address> parameter restricts the display to the entries for the specified IPv6 address. You must specify the <ipv6-address> parameter in hexadecimal using 16-bit values between colons as documented in RFC 2373.

The <ipv6-prefix>/<prefix-length> parameters restrict the display to the entries for the specified IPv6 prefix. You must specify the <ipv6-prefix> parameter in hexadecimal using 16-bit values between colons as documented in RFC 2373. You must specify the <prefix-length> parameter as a decimal value. A slash mark (/) must follow the <ipv6-prefix> parameter and precede the <prefix-length> parameter.

The **connect** keyword restricts the display to entries for directly connected interface IPv6 routes.

The **ospf** keyword restricts the display to entries for OSPFv3 routes.

The **rip** keyword restricts the display to entries for RIPng routes.

The **static** keyword restricts the display to entries for static IPv6 routes.

The **summary** keyword displays a summary of the prefixes and different route types.

The following table lists the information displayed by the **show ipv6 route** command.

Table 28.11:IPv6 route table fields

This Field...	Displays...
Number of entries	The number of entries in the IPv6 route table.
Type	The route type, which can be one of the following: • C – The destination is directly connected to the router. • S – The route is a static route. • R – The route is learned from RIPng. • O – The route is learned from OSPFv3.
IPv6 Prefix	The destination network of the route.
Next-Hop Router	The next-hop router.
Interface	The interface through which this router sends packets to reach the route's destination.
Dis/Metric	The route’s administrative distance and metric value.

To display a summary of the IPv6 route table, enter the following command at any CLI level:

```
FastIron# show ipv6 route summary
IPv6 Routing Table - 7 entries:
  4 connected, 2 static, 0 RIP, 1 OSPF, 0 BGP
  Number of prefixes:
  /16: 1 /32: 1 /64: 3 /128: 2
```

The following table lists the information displayed by the **show ipv6 route summary** command:

Table 28.12:IPv6 route table summary fields

This Field...	Displays...
Number of entries	The number of entries in the IPv6 route table.
Number of route types	The number of entries for each route type.
Number of prefixes	A summary of prefixes in the IPv6 route table, sorted by prefix length.

Displaying Local IPv6 Routers

The Foundry device can function as an IPv6 host, instead of an IPv6 router, if you configure IPv6 addresses on its interfaces but don’t enable IPv6 routing using the **ipv6 unicast-routing** command.

From the IPv6 host, you can display information about IPv6 routers to which the host is connected. The host learns about the routers through their router advertisement messages. To display information about the IPv6 routers connected to an IPv6 host, enter the following command at any CLI level:

```
FastIron# show ipv6 router
Router fe80::2e0:80ff:fe46:3431 on Ethernet 50, last update 0 min
Hops 64, Lifetime 1800 sec
Reachable time 0 msec, Retransmit time 0 msec
```

Syntax: show ipv6 router

If you configure your Foundry device to function as an IPv6 router (you configure IPv6 addresses on its interfaces and enable IPv6 routing using the **ipv6 unicast-routing** command) and you enter the **show ipv6 router command**, you will receive the following output:

```
No IPv6 router in table
```

Meaningful output for this command is generated for Foundry devices configured to function as IPv6 hosts only.

This display shows the following information:

Table 28.13:IPv6 local router information fields

This Field...	Displays...
Router <ipv6 address> on <interface> <port>	The IPv6 address for a particular router interface.
Last update	The amount of elapsed time (in minutes) between the current and previous updates received from a router.
Hops	The default value that should be included in the Hop Count field of the IPv6 header for outgoing IPv6 packets. The hops value applies to the router for which you are displaying information and should be followed by IPv6 hosts attached to the router. A value of 0 indicates that the router leaves this field unspecified.
Lifetime	The amount of time (in seconds) that the router is useful as the default router.
Reachable time	The amount of time (in milliseconds) that a router assumes a neighbor is reachable after receiving a reachability confirmation. The reachable time value applies to the router for which you are displaying information and should be followed by IPv6 hosts attached to the router. A value of 0 indicates that the router leaves this field unspecified.
Retransmit time	The amount of time (in milliseconds) between retransmissions of neighbor solicitation messages. The retransmit time value applies to the router for which you are displaying information and should be followed by IPv6 hosts attached to the router. A value of 0 indicates that the router leaves this field unspecified.

Displaying IPv6 TCP Information

You can display the following IPv6 TCP information:

- General information about each TCP connection on the router, including the percentage of free memory for each of the internal TCP buffers.
- Detailed information about a specified TCP connection.

To display general information about each TCP connection on the router, enter the following command at any CLI level:

```
FastIron# show ipv6 tcp connections
Local IP address:port <-> Remote IP address:port TCP state
192.168.182.110:23 <->    192.168.8.186:4933     ESTABLISHED
192.168.182.110:8218 <->  192.168.182.106:179    ESTABLISHED
192.168.182.110:8039 <->  192.168.2.119:179      SYN-SENT
192.168.182.110:8159 <->  192.168.2.102:179      SYN-SENT
2000:4::110:179 <->       2000:4::106:8222       ESTABLISHED (1440)
Total 5 TCP connections

TCP MEMORY USAGE PERCENTAGE
FREE TCB = 98 percent
FREE TCP QUEUE BUFFER = 99 percent
FREE TCP SEND BUFFER = 97 percent
FREE TCP RECEIVE BUFFER = 100 percent
FREE TCP OUT OF SEQUENCE BUFFER = 100 percent
```

Syntax: show ipv6 tcp connections

This display shows the following information:

Table 28.14:General IPv6 TCP connection fields

This Field...	Displays...
Local IP address:port	The IPv4 or IPv6 address and port number of the local router interface over which the TCP connection occurs.
Remote IP address:port	The IPv4 or IPv6 address and port number of the remote router interface over which the TCP connection occurs.

Table 28.14:General IPv6 TCP connection fields (Continued)

This Field...	Displays...
TCP state	The state of the TCP connection. Possible states include the following: • LISTEN – Waiting for a connection request. • SYN-SENT – Waiting for a matching connection request after having sent a connection request. • SYN-RECEIVED – Waiting for a confirming connection request acknowledgment after having both received and sent a connection request. • ESTABLISHED – Data can be sent and received over the connection. This is the normal operational state of the connection. • FIN-WAIT-1 – Waiting for a connection termination request from the remote TCP, or an acknowledgment of the connection termination request previously sent. • FIN-WAIT-2 – Waiting for a connection termination request from the remote TCP. • CLOSE-WAIT – Waiting for a connection termination request from the local user. • CLOSING – Waiting for a connection termination request acknowledgment from the remote TCP. • LAST-ACK – Waiting for an acknowledgment of the connection termination request previously sent to the remote TCP (which includes an acknowledgment of its connection termination request). • TIME-WAIT – Waiting for enough time to pass to be sure the remote TCP received the acknowledgment of its connection termination request. • CLOSED – There is no connection state.
FREE TCB = <percentage>	The percentage of free TCP control block (TCB) space.
FREE TCB QUEUE BUFFER = <percentage>	The percentage of free TCB queue buffer space.
FREE TCB SEND BUFFER = <percentage>	The percentage of free TCB send buffer space.
FREE TCB RECEIVE BUFFER = <percentage>	The percentage of free TCB receive buffer space.
FREE TCB OUT OF SEQUENCE BUFFER = <percentage>	The percentage of free TCB out of sequence buffer space.

To display detailed information about a specified TCP connection, enter a command such as the following at any CLI level:

```
FastIron# show ipv6 tcp status 2000:4::110 179 2000:4::106 8222
TCP: TCB = 0x217fc300
TCP: 2000:4::110:179 <-> 2000:4::106:8222: state: ESTABLISHED Port: 1
  Send: initial sequence number = 242365900
  Send: first unacknowledged sequence number = 242434080
  Send: current send pointer = 242434080
  Send: next sequence number to send = 242434080
  Send: remote received window = 16384
  Send: total unacknowledged sequence number = 0
  Send: total used buffers 0
  Receive: initial incoming sequence number = 740437769
  Receive: expected incoming sequence number = 740507227
  Receive: received window = 16384
  Receive: bytes in receive queue = 0
  Receive: congestion window = 1459
```

Syntax: show ipv6 tcp status <local-ip-address> <local-port-number> <remote-ip-address> <remote-port-number>

The <local-ip-address> parameter can be the IPv4 or IPv6 address of the local interface over which the TCP connection is taking place.

The <local-port-number> parameter is the local port number over which a TCP connection is taking place.

The <remote-ip-address> parameter can be the IPv4 or IPv6 address of the remote interface over which the TCP connection is taking place.

The <remote-port-number> parameter is the local port number over which a TCP connection is taking place.

This display shows the following information:

Table 28.15:Specific IPv6 TCP connection fields

This Field...	Displays...
TCB = <location>	The location of the TCB.
<local-ip-address> <local-port-number> <remote-ip-address> <remote-port-number> <state> <port>	This field provides a general summary of the following: • The local IPv4 or IPv6 address and port number. • The remote IPv4 or IPv6 address and port number. • The state of the TCP connection. For information on possible states, see Table on page 28-46. • The port numbers of the local interface.
Send: initial sequence number = <number>	The initial sequence number sent by the local router.
Send: first unacknowledged sequence number = <number>	The first unacknowledged sequence number sent by the local router.

Table 28.15:Specific IPv6 TCP connection fields (Continued)

This Field...	Displays...
Send: current send pointer = <number>	The current send pointer.
Send: next sequence number to send = <number>	The next sequence number sent by the local router.
Send: remote received window = <number>	The size of the remote received window.
Send: total unacknowledged sequence number = <number>	The total number of unacknowledged sequence numbers sent by the local router.
Send: total used buffers <number>	The total number of buffers used by the local router in setting up the TCP connection.
Receive: initial incoming sequence number = <number>	The initial incoming sequence number received by the local router.
Receive: expected incoming sequence number = <number>	The incoming sequence number expected by the local router.
Receive: received window = <number>	The size of the local router’s receive window.
Receive: bytes in receive queue = <number>	The number of bytes in the local router’s receive queue.
Receive: congestion window = <number>	The size of the local router’s receive congestion window.

Displaying IPv6 Traffic Statistics

To display IPv6 traffic statistics, enter the following command at any CLI level:

```
FastIron# show ipv6 traffic
IP6 Statistics
  36947 received, 66818 sent, 0 forwarded, 36867 delivered, 0 rawout
  0 bad vers, 23 bad scope, 0 bad options, 0 too many hdr
  0 no route, 0 can't forward, 0 redirect sent
  0 frag recv, 0 frag dropped, 0 frag timeout, 0 frag overflow
  0 reassembled, 0 fragmented, 0 ofragments, 0 can't frag
  0 too short, 0 too small, 11 not member
  0 no buffer, 66819 allocated, 21769 freed
  0 forward cache hit, 46 forward cache miss

ICMP6 Statistics
Received:
  0 dest unreach, 0 pkt too big, 0 time exceeded, 0 param prob
  2 echo req, 1 echo reply, 0 mem query, 0 mem report, 0 mem red
  0 router soli, 2393 router adv, 106 nei soli, 3700 nei adv, 0 redirect
  0 bad code, 0 too short, 0 bad checksum, 0 bad len
  0 reflect, 0 nd toomany opt, 0 badhopcount
Sent:
  0 dest unreach, 0 pkt too big, 0 time exceeded, 0 param prob
  1 echo req, 2 echo reply, 0 mem query, 0 mem report, 0 mem red
  0 router soli, 2423 router adv, 3754 nei soli, 102 nei adv, 0 redirect
  0 error, 0 can't send error, 0 too freq
Sent Errors:
  0 unreach no route, 0 admin, 0 beyond scope, 0 address, 0 no port
  0 pkt too big, 0 time exceed transit, 0 time exceed reassembly
  0 param problem header, 0 nextheader, 0 option, 0 redirect, 0 unknown

UDP Statistics
  470 received, 7851 sent, 6 no port, 0 input errors

TCP Statistics
  57913 active opens, 0 passive opens, 57882 failed attempts
  159 active resets, 0 passive resets, 0 input errors
  565189 in segments, 618152 out segments, 171337 retransmission
```

Syntax: show ipv6 traffic

This display shows the following information:

Table 28.16:IPv6 traffic statistics fields

This Field...	Displays...
IPv6 statistics	
received	The total number of IPv6 packets received by the router.
sent	The total number of IPv6 packets originated and sent by the router.
forwarded	The total number of IPv6 packets received by the router and forwarded to other routers.

Table 28.16:IPv6 traffic statistics fields (Continued)

This Field...	Displays...
delivered	The total number of IPv6 packets delivered to the upper layer protocol.
rawout	This information is used by Foundry Technical Support.
bad vers	The number of IPv6 packets dropped by the router because the version number is not 6.
bad scope	The number of IPv6 packets dropped by the router because of a bad address scope.
bad options	The number of IPv6 packets dropped by the router because of bad options.
too many hdr	The number of IPv6 packets dropped by the router because the packets had too many headers.
no route	The number of IPv6 packets dropped by the router because there was no route.
can’t forward	The number of IPv6 packets the router could not forward to another router.
redirect sent	This information is used by Foundry Technical Support.
frag recv	The number of fragments received by the router.
frag dropped	The number of fragments dropped by the router.
frag timeout	The number of fragment timeouts that occurred.
frag overflow	The number of fragment overflows that occurred.
reassembled	The number of fragmented IPv6 packets that the router reassembled.
fragmented	The number of IPv6 packets fragmented by the router to accommodate the MTU of this router or of another device.
ofragments	The number of output fragments generated by the router.
can’t frag	The number of IPv6 packets the router could not fragment.
too short	The number of IPv6 packets dropped because they are too short.
too small	The number of IPv6 packets dropped because they don’t have enough data.
not member	The number of IPv6 packets dropped because the recipient is not a member of a multicast group.
no buffer	The number of IPv6 packets dropped because there is no buffer available.
forward cache miss	The number of IPv6 packets received for which there is no corresponding cache entry.
ICMP6 statistics Some ICMP statistics apply to both Received and Sent, some apply to Received only, some apply to Sent only, and some apply to Sent Errors only.	
Applies to Received and Sent	
dest unreach	The number of Destination Unreachable messages sent or received by the router.
pkt too big	The number of Packet Too Big messages sent or received by the router.

Table 28.16:IPv6 traffic statistics fields (Continued)

This Field...	Displays...
time exceeded	The number of Time Exceeded messages sent or received by the router.
param prob	The number of Parameter Problem messages sent or received by the router.
echo req	The number of Echo Request messages sent or received by the router.
echo reply	The number of Echo Reply messages sent or received by the router.
mem query	The number of Group Membership Query messages sent or received by the router.
mem report	The number of Membership Report messages sent or received by the router.
mem red	The number of Membership Reduction messages sent or received by the router.
router soli	The number of Router Solicitation messages sent or received by the router.
router adv	The number of Router Advertisement messages sent or received by the router.
nei soli	The number of Neighbor Solicitation messages sent or received by the router.
nei adv	The number of Router Advertisement messages sent or received by the router.
redirect	The number of redirect messages sent or received by the router.
Applies to Received Only	
bad code	The number of Bad Code messages received by the router.
too short	The number of Too Short messages received by the router.
bad checksum	The number of Bad Checksum messages received by the router.
bad len	The number of Bad Length messages received by the router.
nd toomany opt	The number of Neighbor Discovery Too Many Options messages received by the router.
badhopcount	The number of Bad Hop Count messages received by the router.
Applies to Sent Only	
error	The number of Error messages sent by the router.
can’t send error	The number of times the node encountered errors in ICMP error messages.
too freq	The number of times the node has exceeded the frequency of sending error messages.
Applies to Sent Errors Only	
unreach no route	The number of Unreachable No Route errors sent by the router.
admin	The number of Admin errors sent by the router.
beyond scope	The number of Beyond Scope errors sent by the router.
address	The number of Address errors sent by the router.
no port	The number of No Port errors sent by the router.

Table 28.16:IPv6 traffic statistics fields (Continued)

This Field...	Displays...
pkt too big	The number of Packet Too Big errors sent by the router.
time exceed transit	The number of Time Exceed Transit errors sent by the router.
time exceed reassembly	The number of Time Exceed Reassembly errors sent by the router.
param problem header	The number of Parameter Problem Header errors sent by the router.
nextheader	The number of Next Header errors sent by the router.
option	The number of Option errors sent by the router.
redirect	The number of Redirect errors sent by the router.
unknown	The number of Unknown errors sent by the router.
UDP statistics	
received	The number of UDP packets received by the router.
sent	The number of UDP packets sent by the router.
no port	The number of UDP packets dropped because the packet did not contain a valid UDP port number.
input errors	This information is used by Foundry Technical Support.
TCP statistics	
active opens	The number of TCP connections opened by the router by sending a TCP SYN to another device.
passive opens	The number of TCP connections opened by the router in response to connection requests (TCP SYNs) received from other devices.
failed attempts	This information is used by Foundry Technical Support.
active resets	The number of TCP connections the router reset by sending a TCP RESET message to the device at the other end of the connection.
passive resets	The number of TCP connections the router reset because the device at the other end of the connection sent a TCP RESET message.
input errors	This information is used by Foundry Technical Support.
in segments	The number of TCP segments received by the router.
out segments	The number of TCP segments sent by the router.
retransmission	The number of segments that the router retransmitted because the retransmission timer for the segment had expired before the device at the other end of the connection had acknowledged receipt of the segment.

Chapter 29
Configuring IP

This chapter describes the Internet Protocol (IP) parameters on Foundry Layer 2 Switches and Layer 3 Switches and how to configure them.

NOTE: References to chassis-based Layer 3 Switches apply to the FastIron SuperX Switch.

NOTE: The terms *Layer 3 Switch* and *router* are used interchangeably in this chapter and mean the same thing.

Basic Configuration

IP is enabled by default. Basic configuration consists of adding IP addresses and, for Layer 3 Switches, enabling a route exchange protocol, such as Routing Information Protocol (RIP).

- If you are configuring a Layer 3 Switch, see “Configuring IP Addresses” on page 29-16 to add IP addresses, then see one or more of the following to enable and configure the route exchange protocols:
  - “Configuring RIP” on page 33-1
  - “Configuring OSPF Version 2 (IPv4)” on page 35-1
  - “Configuring BGP4” on page 38-1
- If you are configuring a Layer 2 Switch, see “Configuring the Management IP Address and Specifying the Default Gateway” on page 29-58 to add an IP address for management access through the network and to specify the default gateway.

The rest of this chapter describes IP and how to configure it in more detail. Use the information in this chapter if you need to change some of the IP parameters from their default values or you want to view configuration information or statistics.

Overview

Foundry Networks Layer 2 Switches and Layer 3 Switches support Internet Protocol (IP) version 4. IP support on Foundry Layer 2 Switches consists of basic services to support management access and access to a default gateway. IP support on Foundry Layer 3 Switches includes all of the following, in addition to a highly configurable implementation of basic IP services including Address Resolution Protocol (ARP), ICMP Router Discovery Protocol (IRDP), and Reverse ARP (RARP):

- Route exchange protocols
  - Routing Information Protocol (RIP)

  - Open Shortest Path First (OSPF)
  - Border Gateway Protocol version 4 (BGP4)
- Multicast protocols
  - Internet Group Membership Protocol (IGMP)
  - Protocol Independent Multicast Dense (PIM-DM)
  - Protocol Independent Multicast Sparse (PIM-SM)
  - Distance Vector Multicast Routing Protocol (DVMRP)
- Router redundancy protocols
  - Virtual Router Redundancy Protocol Extended (VRRPE)
  - Virtual Router Redundancy Protocol (VRRP)

IP Interfaces

Foundry Layer 3 Switches and Layer 2 Switches allow you to configure IP addresses. On Layer 3 Switches, IP addresses are associated with individual interfaces. On Layer 2 Switches, a single IP address serves as the management access address for the entire device.

All Foundry Layer 3 Switches and Layer 2 Switches support configuration and display of IP address in classical subnet format (example: 192.168.1.1 255.255.255.0) and Classless Interdomain Routing (CIDR) format (example: 192.168.1.1/24). You can use either format when configuring IP address information. IP addresses are displayed in classical subnet format by default but you can change the display format to CIDR. See “Changing the Network Mask Display to Prefix Format” on page 29-64.

Layer 3 Switches

Foundry Layer 3 Switches allow you to configure IP addresses on the following types of interfaces:

- Ethernet ports
- Virtual routing interfaces (used by VLANs to route among one another)
- Loopback interfaces

Each IP address on a Layer 3 Switch must be in a different subnet. You can have only one interface that is in a given subnet. For example, you can configure IP addresses 192.168.1.1/24 and 192.168.2.1/24 on the same Layer 3 Switch, but you cannot configure 192.168.1.1/24 and 192.168.1.2/24 on the same Layer 3 Switch.

You can configure multiple IP addresses on the same interface.

The number of IP addresses you can configure on an individual interface depends on the Layer 3 Switch model. To display the maximum number of IP addresses and other system parameters you can configure on a Layer 3 Switch, see the section “Displaying and Modifying System Parameter Default Settings” on page 8-11.

You can use any of the IP addresses you configure on the Layer 3 Switch for Telnet, Web management, or SNMP access.

Layer 2 Switches

You can configure an IP address on a Foundry Layer 2 Switch for management access to the Layer 2 Switch. An IP address is required for Telnet access, Web management access, and SNMP access.

You also can specify the default gateway for forwarding traffic to other subnets.

IP Packet Flow Through a Layer 3 Switch

Figure 29.1 shows how an IP packet moves through a Foundry Layer 3 Switch.

Figure 29.1 IP Packet Flow through aFoundry Layer 3 Switch

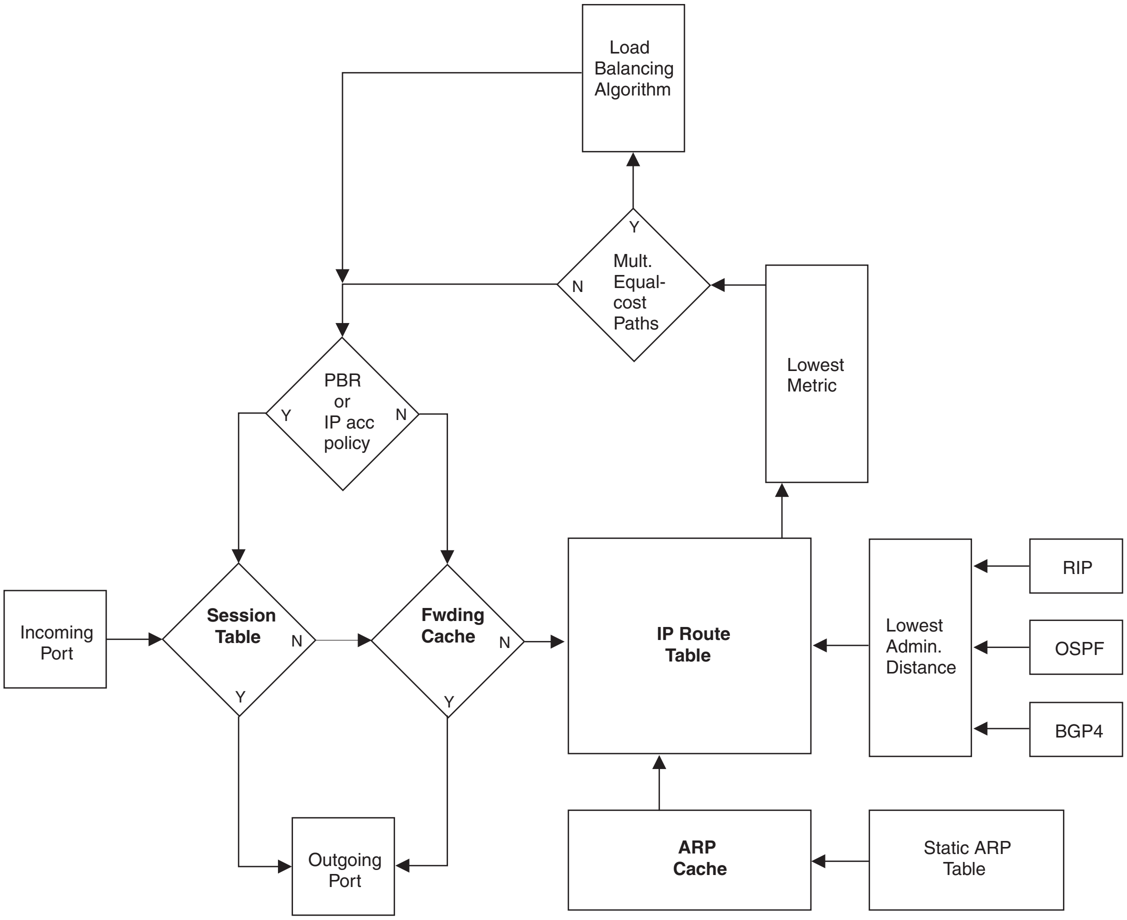


Figure 29.1 shows the following packet flow:

1. When the Layer 3 Switch receives an IP packet, the Layer 3 Switch checks for filters on the receiving interface.[1] If a deny filter on the interface denies the packet, the Layer 3 Switch discards the packet and performs no further processing, except generating a Syslog entry and SNMP message, if logging is enabled for the filter.
2. If the packet is not denied at the incoming interface, the Layer 3 Switch looks in the session table for an entry that has the same source IP address and TCP or UDP port as the packet. If the session table contains a matching entry, the Layer 3 Switch immediately forwards the packet, by addressing it to the destination IP address and TCP or UDP port listed in the session table entry and sending the packet to a queue on the outgoing port(s) listed in the session table. The Layer 3 Switch selects the queue based on the Quality of Service (QoS) level associated with the session table entry.
3. If the session table does not contain an entry that matches the packet’s source address and TCP or UDP port, the Layer 3 Switch looks in the IP forwarding cache for an entry that matches the packet’s destination IP address. If the forwarding cache contains a matching entry, the Layer 3 Switch forwards the packet to the IP address in the entry. The Layer 3 Switch sends the packet to a queue on the outgoing port(s) listed in the forwarding cache. The Layer 3 Switch selects the queue based on the Quality of Service (QoS) level associated with the forwarding cache entry.
4. If the IP forwarding cache does not have an entry for the packet, the Layer 3 Switch checks the IP route table for a route to the packet’s destination. If the IP route table has a route, the Layer 3 Switch makes an entry in the session table or the forwarding cache, and sends the route to a queue on the outgoing port(s).

---

[1] The filter can be an Access Control List (ACL) or an IP access policy.

- If the running-config contains an IP access policy for the packet, the software makes an entry in the session table. The Layer 3 Switch uses the new session table entry to forward subsequent packets from the same source to the same destination.
- If the running-config does not contain an IP access policy for the packet, the software creates a new entry in the forwarding cache. The Layer 3 Switch uses the new cache entry to forward subsequent packets to the same destination.

The following sections describe the IP tables and caches:

- ARP cache and static ARP table
- IP route table
- IP forwarding cache
- IP session table

The software enables you to display these tables. You also can change the capacity of the tables on an individual basis if needed by changing the memory allocation for the table.

ARP Cache and Static ARP Table

The ARP cache contains entries that map IP addresses to MAC addresses. Generally, the entries are for devices that are directly attached to the Layer 3 Switch.

An exception is an ARP entry for an interface-based static IP route that goes to a destination that is one or more router hops away. For this type of entry, the MAC address is either the destination device's MAC address or the MAC address of the router interface that answered an ARP request on behalf of the device, using proxy ARP.

ARP Cache

The ARP cache can contain dynamic (learned) entries and static (user-configured) entries. The software places a dynamic entry in the ARP cache when the Layer 3 Switch learns a device's MAC address from an ARP request or ARP reply from the device.

The software can learn an entry when the Layer 2 Switch or Layer 3 Switch receives an ARP request from another IP forwarding device or an ARP reply. Here is an example of a dynamic entry:

```
      IP Address          MAC Address         Type        Age     Port
1     207.95.6.102        0800.5afc.ea21      Dynamic     0         6
```

Each entry contains the destination device's IP address and MAC address.

Static ARP Table

In addition to the ARP cache, Layer 3 Switches have a static ARP table. Entries in the static ARP table are user-configured. You can add entries to the static ARP table regardless of whether the device the entry is for is connected to the Layer 3 Switch.

NOTE: The Layer 3 Switches have a static ARP table but Layer 2 Switches do not.

The software places an entry from the static ARP table into the ARP cache when the entry's interface comes up.

Here is an example of a static ARP entry:

```
Index   IP Address          MAC Address         Port
1       207.95.6.111        0800.093b.d210      1/1
```

Each entry lists the information you specified when you created the entry.

To display ARP entries, see the following:

- "Displaying the ARP Cache" on page 29-70 – Layer 3 Switch
- "Displaying the Static ARP Table" on page 29-71 – Layer 3 Switch only
- "Displaying ARP Entries" on page 29-80 – Layer 2 Switch

To configure other ARP parameters, see the following:

- “Configuring ARP Parameters” on page 29-26 – Layer 3 Switch only

To increase the size of the ARP cache and static ARP table, see the following:

- For dynamic entries, see the section “Displaying and Modifying System Parameter Default Settings” on page 8-11. The ip-arp parameter controls the ARP cache size.
- Static entries, “Changing the Maximum Number of Entries the Static ARP Table Can Hold” on page 29-30 – Layer 3 Switches only. The ip-static-arp parameter controls the static ARP table size.

IP Route Table

The IP route table contains paths to IP destinations.

NOTE: Layer 2 Switches do not have an IP route table. A Layer 2 Switch sends all packets addressed to another subnet to the default gateway, which you specify when you configure the basic IP information on the Layer 2 Switch.

The IP route table can receive the paths from the following sources:

- A directly-connected destination, which means there are no router hops to the destination
- A static IP route, which is a user-configured route
- A route learned through RIP
- A route learned through OSPF
- A route learned through BGP4

The IP route table contains the best path to a destination.

- When the software receives paths from more than one of the sources listed above, the software compares the administrative distance of each path and selects the path with the lowest administrative distance. The administrative distance is a protocol-independent value from 1 – 255.
- When the software receives two or more best paths from the same source and the paths have the same metric (cost), the software can load share traffic among the paths based on destination host or network address (based on the configuration and the Layer 3 Switch model).

Here is an example of an entry in the IP route table:

```
Destination       NetMask           Gateway           Port   Cost   Type

1.1.0.0           255.255.0.0       99.1.1.2          1/1    2      R
```

Each IP route table entry contains the destination’s IP address and subnet mask and the IP address of the next-hop router interface to the destination. Each entry also indicates the port attached to the destination or the next-hop to the destination, the route’s IP metric (cost), and the type. The type indicates how the IP route table received the route.

To display the IP route table, see the following:

- “Displaying the IP Route Table” on page 29-73 – Layer 3 Switch only

To configure a static IP route, see the following:

- “Configuring Static Routes” on page 29-35 – Layer 3 Switch only

To clear a route from the IP route table, see the following:

- “Clearing IP Routes” on page 29-76 – Layer 3 Switch only

To increase the size of the IP route table for learned and static routes, see the section “Displaying and Modifying System Parameter Default Settings” on page 8-11.

- For learned routes, modify the ip-route parameter.

- For static routes, modify the ip-static-route parameter.

IP Forwarding Cache

The IP forwarding cache provides a fast-path mechanism for forwarding IP packets. The cache contains entries for IP destinations. When a Foundry Layer 3 Switch has completed processing and addressing for a packet and is ready to forward the packet, the device checks the IP forwarding cache for an entry to the packet's destination.

- If the cache contains an entry with the destination IP address, the device uses the information in the entry to forward the packet out the ports listed in the entry. The destination IP address is the address of the packet's final destination. The port numbers are the ports through which the destination can be reached.
- If the cache does not contain an entry and the traffic does not qualify for an entry in the session table instead, the software can create an entry in the forwarding cache.

Each entry in the IP forwarding cache has an age timer. If the entry remains unused for ten minutes, the software removes the entry. The age timer is not configurable.

Here is an example of an entry in the IP forwarding cache:

```
      IP Address       Next Hop         MAC              Type  Port  Vlan  Pri
1     192.168.1.11     DIRECT           0000.0000.0000   PU    n/a         0
```

Each IP forwarding cache entry contains the IP address of the destination, and the IP address and MAC address of the next-hop router interface to the destination. If the destination is actually an interface configured on the Layer 3 Switch itself, as shown here, then next-hop information indicates this. The port through which the destination is reached is also listed, as well as the VLAN and Layer 4 QoS priority associated with the destination if applicable.

To display the IP forwarding cache, see "Displaying the Forwarding Cache" on page 29-72.

NOTE: You cannot add static entries to the IP forwarding cache, although you can increase the number of entries the cache can contain. See the section "Displaying and Modifying System Parameter Default Settings" on page 8-11.

To increase the size of the IP forwarding cache, see the section "Displaying and Modifying System Parameter Default Settings" on page 8-11.

Layer 4 Session Table

The Layer 4 session provides a fast path for forwarding packets. A ***session*** is an entry that contains complete Layer 3 and Layer 4 information for a flow of traffic. Layer 3 information includes the source and destination IP addresses. Layer 4 information includes the source and destination TCP and UDP ports. For comparison, the IP forwarding cache contains the Layer 3 destination address but does not contain the other source and destination address information of a Layer 4 session table entry.

The Layer 2 Switch or Layer 3 Switch selects the session table instead of the IP forwarding table for fast-path forwarding for the following features:

- Layer 4 Quality-of-Service (QoS) policies
- IP access policies

To increase the size of the session table, see the section "Displaying and Modifying System Parameter Default Settings" on page 8-11. The ip-qos-session parameter controls the size of the session table.

IP Route Exchange Protocols

Foundry Layer 3 Switches support the following IP route exchange protocols:

- Routing Information Protocol (RIP)
- Open Shortest Path First (OSPF)
- Border Gateway Protocol version 4 (BGP4)

All these protocols provide routes to the IP route table. You can use one or more of these protocols, in any combination. The protocols are disabled by default. For configuration information, see the following:

- "Configuring RIP" on page 33-1
- "Configuring OSPF Version 2 (IPv4)" on page 35-1
- "Configuring BGP4" on page 38-1

IP Multicast Protocols

Foundry Layer 3 Switches also support the following Internet Group Membership Protocol (IGMP) based IP multicast protocols:

- Protocol Independent Multicast – Dense mode (PIM-DM)
- Protocol Independent Multicast – Sparse mode (PIM-SM)
- Distance Vector Multicast Routing Protocol (DVMRP)

For configuration information, see "Configuring IP Multicast Protocols" on page 30-1.

NOTE: Foundry Layer 2 Switches support IGMP and can forward IP multicast packets. See the chapter "Configuring IP Multicast Traffic Reduction for the FastIron X Series Switch" .

IP Interface Redundancy Protocols

You can configure a Foundry Layer 3 Switch to back up an IP interface configured on another Foundry Layer 3 Switch. If the link for the backed up interface becomes unavailable, the other Layer 3 Switch can continue service for the interface. This feature is especially useful for providing a backup to a network's default gateway.

Foundry Layer 3 Switches support the following IP interface redundancy protocols:

- Virtual Router Redundancy Protocol (VRRP) – A standard router redundancy protocol based on RFC 2338. You can use VRRP to configure Foundry Layer 3 Switches and third-party routers to back up IP interfaces on other Foundry Layer 3 Switches or third-party routers.
- Virtual Router Redundancy Protocol Extended (VRRPE) – A Foundry extension to standard VRRP that adds additional features and overcomes limitations in standard VRRP. You can use VRRPE only on Foundry Layer 3 Switches.

For configuration information, see the following:

  - Virtual Router Redundancy Protocol Extended (VRRPE) – see "Configuring VRRP and VRRPE" on page 37-1.
  - Virtual Router Redundancy Protocol (VRRP) – see "Configuring VRRP and VRRPE" on page 37-1.

Access Control Lists and IP Access Policies

Foundry Layer 3 Switches provide two mechanisms for filtering IP traffic:

- Access Control Lists (ACLs)
- IP access policies

Both methods allow you to filter packets based on Layer 3 and Layer 4 source and destination information.

ACLs also provide great flexibility by providing the input to various other filtering mechanisms such as route maps, which are used by BGP4.

IP access policies allow you to configure QoS based on sessions (Layer 4 traffic flows).

Only one of these filtering mechanisms can be enabled on a Foundry device at a time. Foundry devices can store forwarding information for both methods of filtering in the session table.

For configuration information, "Configuring Rule-Based IP Access Control Lists (ACLs)" on page 17-1

Basic IP Parameters and Defaults – Layer 3 Switches

IP is enabled by default. The following IP-based protocols are all disabled by default:

- Routing protocols
    - Routing Information Protocol (RIP) – see “Configuring RIP” on page 33-1
    - Open Shortest Path First (OSPF) – see “Configuring OSPF Version 2 (IPv4)” on page 35-1
    - Border Gateway Protocol version 4 (BGP4) – see “Configuring BGP4” on page 38-1
- Multicast protocols
    - Internet Group Membership Protocol (IGMP) – see “Changing Global IP Multicast Parameters” on page 30-2
    - Protocol Independent Multicast Dense (PIM-DM) – see “PIM Dense” on page 30-5
    - Protocol Independent Multicast Sparse (PIM-SM) – see “PIM Sparse” on page 30-12
    - Distance Vector Multicast Routing Protocol (DVMRP) – see “DVMRP Overview” on page 30-32
- Router redundancy protocols
    - Virtual Router Redundancy Protocol Extended (VRRPE) – see “Configuring VRRP and VRRPE” on page 37-1.
    - Virtual Router Redundancy Protocol (VRRP) – see “Configuring VRRP and VRRPE” on page 37-1.

The following tables list the Layer 3 Switch IP parameters, their default values, and where to find configuration information.

NOTE: For information about parameters in other protocols based on IP, such as RIP, OSPF, and so on, see the configuration chapters for those protocols.

When Parameter Changes Take Effect

Most IP parameters described in this chapter are dynamic. They take effect immediately, as soon as you enter the CLI command or select the Web management interface option. You can verify that a dynamic change has taken effect by displaying the running-config. To display the running-config, enter the **show running-config** or **write terminal** command at any CLI prompt. (You cannot display the running-config from the Web management interface.)

To save a configuration change permanently so that the change remains in effect following a system reset or software reload, save the change to the startup-config file.

- To save configuration changes to the startup-config file, enter the **write memory** command from the Privileged EXEC level of any configuration level of the CLI.
- To save the configuration changes using the Web management interface, select the Save link at the bottom of the dialog. Select Yes when prompted to save the configuration change to the startup-config file on the device’s flash memory. You also can access the dialog for saving configuration changes by clicking on Command in the tree view, then clicking on Save to Flash.

Changes to memory allocation require you to reload the software after you save the changes to the startup-config file. When reloading the software is required to complete a configuration change described in this chapter, the procedure that describes the configuration change includes a step for reloading the software.

IP Global Parameters – Layer 3 Switches

Table 29.1 lists the IP global parameters for Layer 3 Switches.

Table 29.1: IP Global Parameters – Layer 3 Switches

Parameter	Description	Default	See page...
IP state	The Internet Protocol, version 4	Enabled **Note**: You cannot disable IP.	n/a
IP address and mask notation	Format for displaying an IP address and its network mask information. You can enable one of the following: • Class-based format; example: 192.168.1.1 255.255.255.0 • Classless Interdomain Routing (CIDR) format; example: 192.168.1.1/24	Class-based **Note**: Changing this parameter affects the display of IP addresses, but you can enter addresses in either format regardless of the display setting.	29-64
Router ID	The value that routers use to identify themselves to other routers when exchanging route information. OSPF and BGP4 use router IDs to identify routers. RIP does not use the router ID.	The IP address configured on the lowest-numbered loopback interface. If no loopback interface is configured, then the lowest-numbered IP address configured on the device.	29-24
Maximum Transmission Unit (MTU)	The maximum length an Ethernet packet can be without being fragmented.	1500 bytes for Ethernet II encapsulation 1492 bytes for SNAP encapsulation	29-22
Address Resolution Protocol (ARP)	A standard IP mechanism that routers use to learn the Media Access Control (MAC) address of a device on the network. The router sends the IP address of a device in the ARP request and receives the device’s MAC address in an ARP reply.	Enabled	29-26

Table 29.1: IP Global Parameters – Layer 3 Switches (Continued)

Parameter	Description	Default	See page...
ARP rate limiting	Lets you specify a maximum number of ARP packets the device will accept each second. If the device receives more ARP packets than you specify, the device drops additional ARP packets for the remainder of the one-second interval.	Disabled	29-27
ARP age	The amount of time the device keeps a MAC address learned through ARP in the device's ARP cache. The device resets the timer to zero each time the ARP entry is refreshed and removes the entry if the timer reaches the ARP age. **Note**: You also can change the ARP age on an individual interface basis. See Table 29.2 on page 29-12.	Ten minutes	29-27
Proxy ARP	An IP mechanism a router can use to answer an ARP request on behalf of a host, by replying with the router's own MAC address instead of the host's.	Disabled	29-28
Static ARP entries	An ARP entry you place in the static ARP table. Static entries do not age out.	No entries	29-29
Time to Live (TTL)	The maximum number of routers (hops) through which a packet can pass before being discarded. Each router decreases a packet's TTL by 1 before forwarding the packet. If decreasing the TTL causes the TTL to be 0, the router drops the packet instead of forwarding it.	64 hops	29-31
Directed broadcast forwarding	A directed broadcast is a packet containing all ones (or in some cases, all zeros) in the host portion of the destination IP address. When a router forwards such a broadcast, it sends a copy of the packet out each of its enabled IP interfaces. **Note**: You also can enable or disable this parameter on an individual interface basis. See Table 29.2 on page 29-12.	Disabled	29-31
Directed broadcast mode	The packet format the router treats as a directed broadcast. The following formats can be directed broadcast: • All ones in the host portion of the packet's destination address. • All zeroes in the host portion of the packet's destination address.	All ones **Note**: If you enable all-zeroes directed broadcasts, all-ones directed broadcasts remain enabled.	29-32
Source-routed packet forwarding	A source-routed packet contains a list of IP addresses through which the packet must pass to reach its destination.	Enabled	29-31
Internet Control Message Protocol (ICMP) messages	The Foundry Layer 3 Switch can send the following types of ICMP messages: • Echo messages (ping messages) • Destination Unreachable messages	Enabled	29-32

Table 29.1: IP Global Parameters – Layer 3 Switches (Continued)

Parameter	Description	Default	See page...
ICMP Router Discovery Protocol (IRDP)	An IP protocol a router can use to advertise the IP addresses of its router interfaces to directly attached hosts. You can enable or disable the protocol, and change the following protocol parameters: • Forwarding method (broadcast or multicast) • Hold time • Maximum advertisement interval • Minimum advertisement interval • Router preference level **Note**: You also can enable or disable IRDP and configure the parameters on an individual interface basis. See Table 29.2 on page 29-12.	Disabled	29-45
Reverse ARP (RARP)	An IP mechanism a host can use to request an IP address from a directly attached router when the host boots.	Enabled	29-47
Static RARP entries	An IP address you place in the RARP table for RARP requests from hosts. **Note**: You must enter the RARP entries manually. The Layer 3 Switch does not have a mechanism for learning or dynamically generating RARP entries.	No entries	29-47
Maximum BootP relay hops	The maximum number of hops away a BootP server can be located from a router and still be used by the router's clients for network booting.	Four	29-51
Domain name for Domain Name Server (DNS) resolver	A domain name (example: foundry.router.com) you can use in place of an IP address for certain operations such as IP pings, trace routes, and Telnet management connections to the router.	None configured	29-19
DNS default gateway addresses	A list of gateways attached to the router through which clients attached to the router can reach DNSs.	None configured	29-19
IP load sharing	A Foundry feature that enables the router to balance traffic to a specific destination across multiple equal-cost paths. On X Series devices, IP load sharing uses a hashing algorithm based on the source IP address, destination IP address, and protocol field in the IP header. **Note**: Load sharing is sometimes called Equal Cost Multi Path (ECMP).	Enabled	29-42
Maximum IP load sharing paths	The maximum number of equal-cost paths across which the Layer 3 Switch is allowed to distribute traffic.	Four	29-45

Table 29.1: IP Global Parameters – Layer 3 Switches (Continued)

Parameter	Description	Default	See page...
Origination of default routes	You can enable a router to originate default routes for the following route exchange protocols, on an individual protocol basis: • RIP • OSPF • BGP4	Disabled	33-7 35-31 38-25
Default network route	The router uses the default network route if the IP route table does not contain a route to the destination and also does not contain an explicit default route (0.0.0.0 0.0.0.0 or 0.0.0.0/0).	None configured	29-41
Static route	An IP route you place in the IP route table.	No entries	29-35
Source interface	The IP address the router uses as the source address for Telnet, RADIUS, or TACACS/TACACS+ packets originated by the router. The router can select the source address based on either of the following: • The lowest-numbered IP address on the interface the packet is sent on. • The lowest-numbered IP address on a specific interface. The address is used as the source for all packets of the specified type regardless of interface the packet is sent on.	The lowest-numbered IP address on the interface the packet is sent on.	29-25

IP Interface Parameters – Layer 3 Switches

Table 29.2 lists the interface-level IP parameters for Layer 3 Switches.

Table 29.2: IP Interface Parameters – Layer 3 Switches

Parameter	Description	Default	See page...
IP state	The Internet Protocol, version 4	Enabled **Note**: You cannot disable IP.	n/a
IP address	A Layer 3 network interface address **Note**: Layer 2 Switches have a single IP address used for management access to the entire device. Layer 3 Switches have separate IP addresses on individual interfaces.	None configured[1]	29-16
Encapsulation type	The format of the packets in which the router encapsulates IP datagrams. The encapsulation format can be one of the following: • Ethernet II • SNAP	Ethernet II	29-22

Table 29.2: IP Interface Parameters – Layer 3 Switches (Continued)

Parameter	Description	Default	See page...
Maximum Transmission Unit (MTU)	The maximum length (number of bytes) of an encapsulated IP datagram the router can forward.	1500 for Ethernet II encapsulated packets 1492 for SNAP encapsulated packets	29-23
ARP age	Locally overrides the global setting. See Table 29.1 on page 29-9.	Ten minutes	29-27
Metric	A numeric cost the router adds to RIP routes learned on the interface. This parameter applies only to RIP routes.	1 (one)	33-4
Directed broadcast forwarding	Locally overrides the global setting. See Table 29.1 on page 29-9.	Disabled	29-31
ICMP Router Discovery Protocol (IRDP)	Locally overrides the global IRDP settings. See Table 29.1 on page 29-9.	Disabled	29-46
DHCP gateway stamp	The router can assist DHCP/BootP Discovery packets from one subnet to reach DHCP/BootP servers on a different subnet by placing the IP address of the router interface that receives the request in the request packet's Gateway field. You can override the default and specify the IP address to use for the Gateway field in the packets. **Note**: UDP broadcast forwarding for client DHCP/BootP requests (bootps) must be enabled (this is enabled by default) and you must configure an IP helper address (the server's IP address or a directed broadcast to the server's subnet) on the port connected to the client.	The lowest-numbered IP address on the interface that receives the request	29-51
DHCP Client-Based Auto-Configuration	Allows the switch to obtain IP addresses from a DHCP host automatically, for either a specified (leased) or infinite period of time.	Enabled	29-52
UDP broadcast forwarding	The router can forward UDP broadcast packets for UDP applications such as BootP. By forwarding the UDP broadcasts, the router enables clients on one subnet to find servers attached to other subnets. **Note**: To completely enable a client's UDP application request to find a server on another subnet, you must configure an IP helper address consisting of the server's IP address or the directed broadcast address for the subnet that contains the server. See the next row.	The router helps forward broadcasts for the following UDP application protocols: • bootps • dns • netbios-dgm • netbios-ns • tacacs • tftp • time	29-49

Table 29.2: IP Interface Parameters – Layer 3 Switches (Continued)

Parameter	Description	Default	See page...
IP helper address	The IP address of a UDP application server (such as a BootP or DHCP server) or a directed broadcast address. IP helper addresses allow the router to forward requests for certain UDP applications from a client on one subnet to a server on another subnet.	None configured	29-50

1. Some devices have a factory default, such as 209.157.22.154, used for troubleshooting during installation. For Layer 3 Switches, the address is on module 1 port 1 (or 1/1).

Basic IP Parameters and Defaults – Layer 2 Switches

IP is enabled by default. The following tables list the Layer 2 Switch IP parameters, their default values, and where to find configuration information.

NOTE: Foundry Layer 2 Switches also provide IP multicast forwarding, which is enabled by default. For information about this feature, see the chapter “Configuring IP Multicast Traffic Reduction for the FastIron X Series Switch” on page 27-1.

IP Global Parameters – Layer 2 Switches

Table 29.3 lists the IP global parameters for Layer 2 Switches.

Table 29.3: IP Global Parameters – Layer 2 Switches

Parameter	Description	Default	See page...
IP address and mask notation	Format for displaying an IP address and its network mask information. You can enable one of the following: • Class-based format; example: 192.168.1.1 255.255.255.0 • Classless Interdomain Routing (CIDR) format; example: 192.168.1.1/24	Class-based **Note**: Changing this parameter affects the display of IP addresses, but you can enter addresses in either format regardless of the display setting.	29-64
IP address	A Layer 3 network interface address **Note**: Layer 2 Switches have a single IP address used for management access to the entire device. Layer 3 Switches have separate IP addresses on individual interfaces.	None configured[1]	29-58
Default gateway	The IP address of a locally attached router (or a router attached to the Layer 2 Switch by bridges or other Layer 2 Switches). The Layer 2 Switch and clients attached to it use the default gateway to communicate with devices on other subnets.	None configured	29-58

Table 29.3: IP Global Parameters – Layer 2 Switches (Continued)

Parameter	Description	Default	See page...
Address Resolution Protocol (ARP)	A standard IP mechanism that networking devices use to learn the Media Access Control (MAC) address of another device on the network. The Layer 2 Switch sends the IP address of a device in the ARP request and receives the device's MAC address in an ARP reply.	Enabled **Note**: You cannot disable ARP.	n/a
ARP age	The amount of time the device keeps a MAC address learned through ARP in the device's ARP cache. The device resets the timer to zero each time the ARP entry is refreshed and removes the entry if the timer reaches the ARP age.	Ten minutes **Note**: You cannot change the ARP age on Layer 2 Switches.	n/a
Time to Live (TTL)	The maximum number of routers (hops) through which a packet can pass before being discarded. Each router decreases a packet's TTL by 1 before forwarding the packet. If decreasing the TTL causes the TTL to be 0, the router drops the packet instead of forwarding it.	64 hops	29-60
Domain name for Domain Name Server (DNS) resolver	A domain name (example: foundry.router.com) you can use in place of an IP address for certain operations such as IP pings, trace routes, and Telnet management connections to the router.	None configured	29-59
DNS default gateway addresses	A list of gateways attached to the router through which clients attached to the router can reach DNSs.	None configured	29-59
Source interface	The IP address the Layer 2 Switch uses as the source address for Telnet, RADIUS, or TACACS/TACACS+ packets originated by the router. The Layer 2 Switch uses its management IP address as the source address for these packets.	The management IP address of the Layer 2 Switch. **Note**: This parameter is not configurable on Layer 2 Switches.	n/a
DHCP gateway stamp	The device can assist DHCP/BootP Discovery packets from one subnet to reach DHCP/BootP servers on a different subnet by placing the IP address of the router interface that forwards the packet in the packet's Gateway field. You can specify up to 32 gateway lists. A gateway list contains up to eight gateway IP addresses. You activate DHCP assistance by associating a gateway list with a port. When you configure multiple IP addresses in a gateway list, the Layer 2 Switch inserts the addresses into the DHCP Discovery packets in a round robin fashion.	None configured	29-63
DHCP Client-Based Auto-Configuration	Allows the switch to obtain IP addresses from a DHCP host automatically, for either a specified (leased) or infinite period of time.	Enabled	29-52

[1] Some devices have a factory default, such as 209.157.22.154, used for troubleshooting during installation. For Layer 3 Switches, the address is on port 1 (or 1/1).

Interface IP Parameters – Layer 2 Switches

Table 29.4 lists the interface-level IP parameters for Layer 2 Switches.

Table 29.4: Interface IP Parameters – Layer 2 Switches

Parameter	Description	Default	See page...
DHCP gateway stamp	You can configure a list of DHCP stamp addresses for a port. When the port receives a DHCP/BootP Discovery packet from a client, the port places the IP address(es) in the gateway list into the packet's Gateway field.	None configured	29-63

Configuring IP Parameters – Layer 3 Switches

The following sections describe how to configure IP parameters. Some parameters can be configured globally while others can be configured on individual interfaces. Some parameters can be configured globally and overridden for individual interfaces.

NOTE: This section describes how to configure IP parameters for Layer 3 Switches. For IP configuration information for Layer 2 Switches, see "Configuring IP Parameters – Layer 2 Switches" on page 29-58.

Configuring IP Addresses

You can configure an IP address on the following types of Layer 3 Switch interfaces:

- Ethernet port
- Virtual routing interface (also called a Virtual Ethernet or "VE")
- Loopback interface

By default, you can configure up to 24 IP addresses on each interface. On Compact Layer 3 Switches, you can increase this amount to up to 64 IP subnet addresses per port by increasing the size of the subnet-per-interface table. See the section "Displaying and Modifying System Parameter Default Settings" on page 8-11.

NOTE: Once you configure a virtual routing interface on a VLAN, you cannot configure Layer 3 interface parameters on individual ports in the VLAN. Instead, you must configure the parameters on the virtual routing interface itself.

Foundry devices support both classical IP network masks (Class A, B, and C subnet masks, and so on) and Classless Interdomain Routing (CIDR) network prefix masks.

- To enter a classical network mask, enter the mask in IP address format. For example, enter "209.157.22.99 255.255.255.0" for an IP address with a Class-C subnet mask.
- To enter a prefix network mask, enter a forward slash (/) and the number of bits in the mask immediately after the IP address. For example, enter "209.157.22.99/24" for an IP address that has a network mask with 24 significant bits (ones).

By default, the CLI displays network masks in classical IP address format (example: 255.255.255.0). You can change the display to prefix format. See "Changing the Network Mask Display to Prefix Format" on page 29-64.

Assigning an IP Address to an Ethernet Port

To assign an IP address to port 1/1, enter the following commands:

```
FastIron(config)#interface ethernet 1/1
FastIron(config-if-1/1)#ip address 192.45.6.1 255.255.255.0
```

NOTE: You also can enter the IP address and mask in CIDR format, as follows:

```
FastIron(config-if-1/1)#ip address 192.45.6.1/24
```

Syntax: [no] ip address <ip-addr> <ip-mask> [ospf-ignore | ospf-passive | secondary]

or

Syntax: [no] ip address <ip-addr>/<mask-bits> [ospf-ignore | ospf-passive | secondary]

The **ospf-ignore** | **ospf-passive** parameters modify the Layer 3 Switch defaults for adjacency formation and interface advertisement. Use one of these parameters if you are configuring multiple IP subnet addresses on the interface but you want to prevent OSPF from running on some of the subnets.

- **ospf-passive** – This option disables adjacency formation with OSPF neighbors. By default, when OSPF is enabled on an interface, the software forms OSPF router adjacencies between each primary IP address on the interface and the OSPF neighbor attached to the interface.
- **ospf-ignore** – This option disables OSPF adjacency formation and also disables advertisement of the interface into OSPF. The subnet is completely ignored by OSPF.

NOTE: The **ospf-passive** option disables adjacency formation but does not disable advertisement of the interface into OSPF. To disable advertisement in addition to disabling adjacency formation, you must use the **ospf-ignore** option.

Use the **secondary** parameter if you have already configured an IP address within the same subnet on the interface.

NOTE: When you configure more than one address in the same subnet, all but the first address are secondary addresses and do not form OSPF adjacencies.

NOTE: Starting in software release FSX 03.0.00, all physical IP interfaces on FastIron X Series Layer 3 devices share the same MAC address. For this reason, if more than one connection is made between two devices, one of which is a FastIron X Series Layer 3 device, Foundry recommends the use of virtual interfaces. It is not recommended to connect two or more physical IP interfaces between two routers.

Assigning an IP Address to a Loopback Interface

Loopback interfaces are always up, regardless of the states of physical interfaces. They can add stability to the network because they are not subject to route flap problems that can occur due to unstable links between a Layer 3 Switch and other devices. You can configure up to eight loopback interfaces on a Chassis Layer 3 Switch and up to four loopback interfaces on a Compact Layer 3 Switch.

You can add up to 24 IP addresses to each loopback interface.

NOTE: If you configure the Foundry Layer 3 Switch to use a loopback interface to communicate with a BGP4 neighbor, you also must configure a loopback interface on the neighbor and configure the neighbor to use that loopback interface to communicate with the Foundry Layer 3 Switch. See “Adding a Loopback Interface” on page 38-10.

To add a loopback interface, enter commands such as those shown in the following example:

```
FastIron(config-bgp-router)#exit
FastIron(config)#int loopback 1
FastIron(config-lbif-1)#ip address 10.0.0.1/24
```

Syntax: interface loopback <num>

The <num> parameter specifies the virtual interface number. You can specify from 1 to the maximum number of virtual interfaces supported on the device. To display the maximum number of virtual interfaces supported on the device, enter the **show default values** command. The maximum is listed in the System Parameters section, in the Current column of the virtual-interface row.

See the syntax description in "Assigning an IP Address to an Ethernet Port" on page 29-17.

Assigning an IP Address to a Virtual Interface

A virtual interface is a logical port associated with a Layer 3 Virtual LAN (VLAN) configured on a Layer 3 Switch. You can configure routing parameters on the virtual interface to enable the Layer 3 Switch to route protocol traffic from one Layer 3 VLAN to the other, without using an external router.[1]

You can configure IP routing interface parameters on a virtual interface. This section describes how to configure an IP address on a virtual interface. Other sections in this chapter that describe how to configure interface parameters also apply to virtual interfaces.

NOTE: The Layer 3 Switch uses the lowest MAC address on the device (the MAC address of port 1 or 1/1) as the MAC address for all ports within all virtual interfaces you configure on the device.

To add a virtual interface to a VLAN and configure an IP address on the interface, enter commands such as the following:

```
FastIron(config)#vlan 2 name IP-Subnet_1.1.2.0/24
FastIron(config-vlan-2)#untag e1 to 4
FastIron(config-vlan-2)#router-interface ve1
FastIron(config-vlan-2)#interface ve1
FastIron(config-vif-1)#ip address 1.1.2.1/24
```

The first two commands in this example create a Layer 3 protocol-based VLAN name "IP-Subnet_1.1.2.0/24" and add a range of untagged ports to the VLAN. The **router-interface** command creates virtual interface 1 as the routing interface for the VLAN. The last two commands change to the interface configuration level for the virtual interface and assign an IP address to the interface.

Syntax: router-interface ve <num>

Syntax: interface ve <num>

See the syntax description in "Assigning an IP Address to an Ethernet Port" on page 29-17.

Configuring IP Follow on a Virtual Routing Interface

Platform Support:

- FESX/FSX/FWSX devices running software release 03.2.00 and later

IP Follow allows multiple virtual routing interfaces to share the same IP address. With this feature, one virtual routing interface is configured with an IP address, while the other virtual routing interfaces are configured to use that IP address, thus, they "follow" the virtual routing interface that has the IP address. This feature is helpful in conserving IP address space.

Configuration Limitations and Feature Limitations

- When configuring IP Follow, the primary virtual routing interface should not have ACL or DoS Protection configured. It is recommended that you create a dummy virtual routing interface as the primary and use the IP-follow virtual routing interface for the network.
- Global Policy Based Routing is not supported when IP Follow is configured.

---

[1] Foundry's feature that allows routing between VLANs within the same device, without the need for external routers, is called Integrated Switch Routing (ISR).

- IPv6 is not supported with **ip-follow**.
- FastIron devices support **ip-follow** with OSPF and VRRP protocols only.

Configuration Syntax

Configure IP Follow by entering commands such as the following:

```
FastIron(config)#vlan 2 name IP-Subnet_1.1.2.0/24
FastIron(config-vlan-2)#untag e1 to 4
FastIron(config-vlan-2)#router-interface ve1
FastIron(config-vlan-2)#interface ve 1
FastIron(config-vif-1)#ip address 10.10.2.1/24
FastIron(config-vif-1)#interface ve 2
FastIron(config-vif-2)#ip follow ve 1
FastIron(config-vif-2)#interface ve 3
FastIron(config-vif-3)#ip follow ve 1
```

Syntax: [no] ip follow ve <number>

For <number> enter the ID of the virtual routing interface.

Use the no form of the command to disable the configuration.

Virtual routing interface 2 and 3 do not have their own IP subnet address, but are sharing the IP address of virtual routing interface 1.

Deleting an IP Address

To delete an IP address, enter a command such as the following:

```
FastIron(config-if-e1000-1)#no ip address 1.1.2.1
```

This command deletes IP address 1.1.2.1. You do not need to enter the subnet mask.

To delete all IP addresses from an interface, enter the following command:

```
FastIron(config-if-e1000-1)#no ip address *
```

Syntax: no ip address <ip-addr> | *

Configuring Domain Name Server (DNS) Resolver

Platform Support:

- FESX/FSX/FWSX devices running software release 03.0.00 and later

The DNS resolver is a feature in a Layer 2 Switch or Layer 3 Switch that sends and receives queries to and from the DNS server on behalf of a client.

- In software releases prior to 03.0.00 for the FastIron X Series devices, you can use one domain name to perform Telnet, ping, traceroute and other DNS query commands. You define one domain name on a Foundry Layer 2 Switch or Layer 3 Switch and up to four DNS servers. Host names and their IP addresses are configured on the DNS servers.
- Starting in software release 03.0.00 for the FastIron X Series devices, you can create a list of domain names that can be used to resolve host names. This list can have more than one domain name. When a client performs a DNS query, all hosts within the domains in the list can be recognized and queries can be sent to any domain on the list.

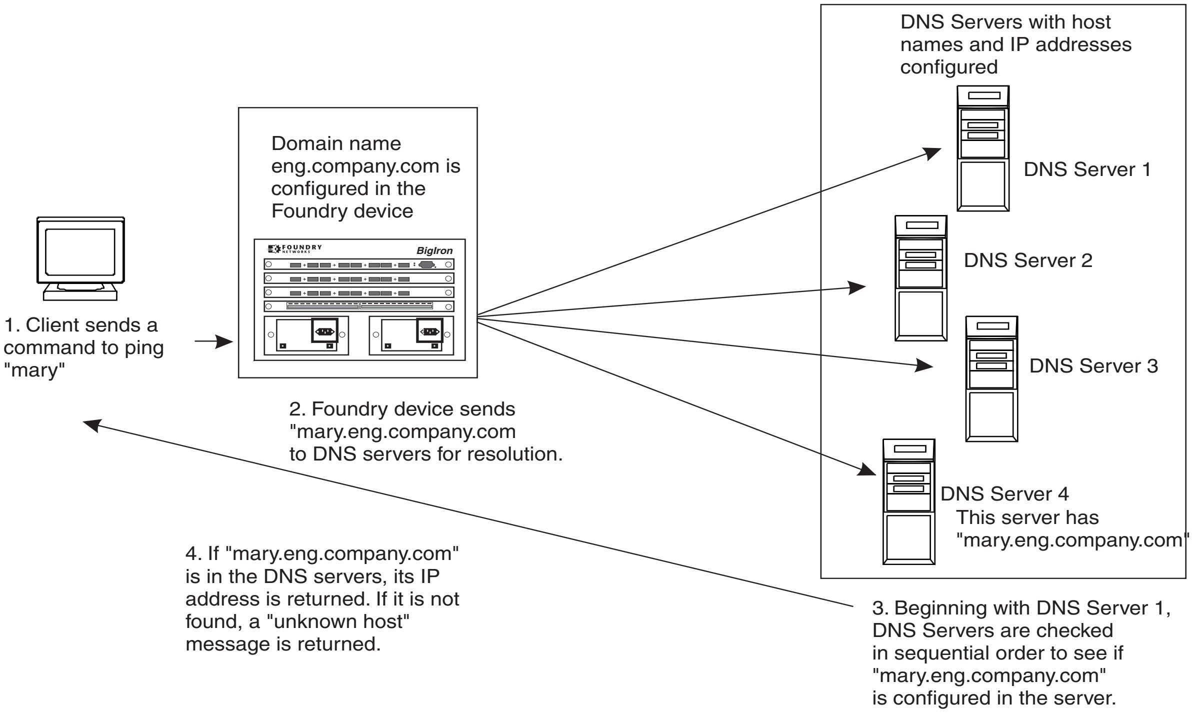
When a client performs a DNS query, all hosts within that domain can be recognized. After you define a domain name, the Foundry device automatically appends the appropriate domain to a host and forwards it to the DNS servers for resolution.

For example, if the domain “ds.company.com” is defined on a Foundry Layer 2 Switch or Layer 3 Switch and you want to initiate a ping to “mary”. You need to reference only the host name instead of the host name and its domain name. For example, you could enter the following command to initiate the ping:

```
U:> ping mary
```

The Foundry Layer 2 Switch or Layer 3 Switch qualifies the host name by appending a domain name. For example, `mary.ds1.company.com`. This qualified name is sent to the DNS server for resolution. If there are four DNS servers configured, it is sent to the first DNS server. If the host name is not resolved, it is sent to the second DNS server. If a match is found, a response is sent back to the client with the host’s IP address. If no match is found, a “unknown host” message is returned. (See Figure 29.2.)

Figure 29.2 DNS Resolution with one Domain Name


DNS Servers with host names and IP addresses configured
Domain name eng.company.com is configured in the Foundry device
FOUNDRY NETWORKS
BigIron
DNS Server 1
DNS Server 2
DNS Server 3
DNS Server 4
This server has "mary.eng.company.com"
1. Client sends a command to ping "mary"
2. Foundry device sends "mary.eng.company.com to DNS servers for resolution.
3. Beginning with DNS Server 1, DNS Servers are checked in sequential order to see if "mary.eng.company.com" is configured in the server.
4. If "mary.eng.company.com" is in the DNS servers, its IP address is returned. If it is not found, a "unknown host" message is returned.

Defining a Domain Name

To define a domain to resolve host names, enter a command such as the following:

```
FastIron(config)#ip dns domain-name ds.company.com
```

Syntax: [no] ip dns domain-name <domain-name>

Enter the domain name for <domain-name>.

Defining DNS Server Addresses

You can configure the Foundry device to recognize up to four DNS servers. The first entry serves as the primary default address. If a query to the primary address fails to be resolved after three attempts, the next DNS address is queried (also up to three times). This process continues for each defined DNS address until the query is resolved. The order in which the default DNS addresses are polled is the same as the order in which you enter them.

To define DNS servers, enter a command such as the following:

```
FastIron(config)#ip dns server-address 209.157.22.199 205.96.7.15 208.95.7.25
201.98.7.15
```

Syntax: [no] ip dns server-address <ip-addr> [<ip-addr>] [<ip-addr>] [<ip-addr>]

In this example, the first IP address entered becomes the primary DNS address and all others are secondary addresses. Because IP address 201.98.7.15 is the last address listed, it is also the last address consulted to resolve a query.

Defining a Domain List

NOTE: Domain lists are supported starting in software release 03.0.00 for the FESX, FSX, FSX 800, FSX 1600, and FWSX.

If you want to use more than one domain name to resolve host names, you can create a list of domain names. For example, enter the commands such as the following:

```
FastIron(config)#ip dns domain-list company.com
FastIron(config)#ip dns domain-list ds.company.com
FastIron(config)#ip dns domain-list hw_company.com
FastIron(config)#ip dns domain-list qa_company.com
FastIron(config)#
```

The domain names are tried in the order you enter them

Syntax: [no] ip dns domain-list <domain-name>

Using a DNS Name to Initiate a Trace Route

Suppose you want to trace the route from a Foundry Layer 3 Switch to a remote server identified as NYC02 on domain newyork.com. Because the NYC02@ds1.newyork.com domain is already defined on the Layer 3 Switch, you need to enter only the host name, NYC02, as noted below.

```
FastIron#traceroute nyc02
```

Syntax: traceroute <host-ip-addr> [maxttl <value>] [minttl <value>] [numeric] [timeout <value>] [source-ip <ip addr>]

The only required parameter is the IP address of the host at the other end of the route.

After you enter the command, a message indicating that the DNS query is in process and the current gateway address (IP address of the domain name server) being queried appear on the screen:

```
Type Control-c to abort
Sending DNS Query to 209.157.22.199
Tracing Route to IP node 209.157.22.80
To ABORT Trace Route, Please use stop-traceroute command.
 Traced route to target IP node 209.157.22.80:
   IP Address          Round Trip Time1    Round Trip Time2
  207.95.6.30         93 msec             121 msec
```

NOTE: In the above example, 209.157.22.199 is the IP address of the domain name server (default DNS gateway address), and 209.157.22.80 represents the IP address of the NYC02 host.

Configuring Packet Parameters

You can configure the following packet parameters on Layer 3 Switches. These parameters control how the Layer 3 Switch sends IP packets to other devices on an Ethernet network. The Layer 3 Switch always places IP packets into Ethernet packets to forward them on an Ethernet port.

- Encapsulation type – The format for the Layer 2 packets within which the Layer 3 Switch sends IP packets.
- Maximum Transmission Unit (MTU) – The maximum length of IP packet that a Layer 2 packet can contain. IP packets that are longer than the MTU are fragmented and sent in multiple Layer 2 packets. You can change the MTU globally or an individual ports.
  - Global MTU – The default MTU value depends on the encapsulation type on a port and is 1500 bytes for Ethernet II encapsulation and 1492 bytes for SNAP encapsulation.
  - Port MTU – A port's default MTU depends on the encapsulation type enabled on the port.

Changing the Encapsulation Type

The Layer 3 Switch encapsulates IP packets into Layer 2 packets, to send the IP packets on the network. (A Layer 2 packet is also called a MAC layer packet or an Ethernet frame.) The source address of a Layer 2 packet is the MAC address of the Layer 3 Switch interface sending the packet. The destination address can be one of the following:

- The MAC address of the IP packet's destination. In this case, the destination device is directly connected to the Layer 3 Switch.
- The MAC address of the next-hop gateway toward the packet's destination.
- An Ethernet broadcast address.

The entire IP packet, including the source and destination address and other control information and the data, is placed in the data portion of the Layer 2 packet. Typically, an Ethernet network uses one of two different formats of Layer 2 packet:

- Ethernet II
- Ethernet SNAP (also called IEEE 802.3)

The control portions of these packets differ slightly. All IP devices on an Ethernet network must use the same format. Foundry Layer 3 Switches use Ethernet II by default. You can change the IP encapsulation to Ethernet SNAP on individual ports if needed.

NOTE: All devices connected to the Layer 3 Switch port must use the same encapsulation type.

To change the IP encapsulation type on interface 5 to Ethernet SNAP, enter the following commands:

```
FastIron(config)#int e 5
FastIron(config-if-e1000-5)#ip encapsulation snap
```

Syntax: ip encapsulation snap | ethernet_ii

Changing the Maximum Transmission Unit (MTU)

The Maximum Transmission Unit (MTU) is the maximum length of IP packet that a Layer 2 packet can contain. IP packets that are longer than the MTU are fragmented and sent in multiple Layer 2 packets. You can change the MTU globally or on individual ports.

The default MTU is 1500 bytes for Ethernet II packets and 1492 for Ethernet SNAP packets.

MTU Enhancements

Foundry devices contain the following enhancements to jumbo packet support:

- Hardware forwarding of Layer 3 jumbo packets – Layer 3 IP unicast jumbo packets received on a port that supports the frame's MTU size and forwarded to another port that also supports the frame's MTU size are forwarded in hardware. Previous releases support hardware forwarding of Layer 2 jumbo frames only.
- ICMP unreachable message if a frame is too large to be forwarded – If a jumbo packet has the Don't Fragment (DF) bit set, and the outbound interface does not support the packet's MTU size, the Foundry device sends an ICMP unreachable message to the device that sent the packet.

NOTE: These enhancements apply only to transit traffic forwarded through the Foundry device.

Configuration Considerations for Increasing the MTU

- When you increase the MTU size of a port, the increase uses system resources. Increase the MTU size only on the ports that need it. For example, if you have one port connected to a server that uses jumbo frames and two other ports connected to clients that can support the jumbo frames, increase the MTU only on those three ports. Leave the MTU size on the other ports at the default value (1500 bytes). Globally increase the MTU size only if needed.
- Use the same MTU size on all ports that will be supporting jumbo frames. If the device needs to fragment a jumbo frame (and the frame does not have the DF bit set), the device fragments the frame into 1500-byte

fragments, even if the outbound port has a larger MTU. For example, if a port has an MTU setting of 8000 and receives an 8000-byte frame, then must forward the frame onto a port with an MTU of 4000, the device does not fragment the 8000-byte frame into two 4000-byte frames. Instead, the device fragments the 8000-byte frame into six fragments (five 1500-byte fragments and a final, smaller fragment.)

Globally Changing the Maximum Transmission Unit (MTU)

The Maximum Transmission Unit (MTU) is the maximum size an IP packet can be when encapsulated in a Layer 2 packet. If an IP packet is larger than the MTU allowed by the Layer 2 packet, the Layer 3 Switch fragments the IP packet into multiple parts that will fit into the Layer 2 packets, and sends the parts of the fragmented IP packet separately, in different Layer 2 packets. The device that receives the multiple fragments of the IP packet reassembles the fragments into the original packet.

You can increase the MTU size to accommodate jumbo packet sizes up to 9216 bytes.

To globally enable jumbo support on all ports of an X Series device, enter commands such as the following:

```
FastIron(config)#jumbo
FastIron(config)#write memory
FastIron(config)#end
FastIron#reload
```

Syntax: [no] jumbo

The above commands configure the Foundry device to forward Ethernet frames that are up to 9216 bytes long.

NOTE: You must save the configuration change and then reload the software to enable jumbo support.

Changing the Maximum Transmission Unit on an Individual Port

By default, the maximum Ethernet MTU sizes are as follows:

- 1500 bytes – The maximum for Ethernet II encapsulation
- 1492 bytes – The maximum for SNAP encapsulation

When jumbo mode is enabled, the maximum Ethernet MTU sizes are as follows:

- 9216 bytes – The maximum for Ethernet II encapsulation
- 9216 bytes – The maximum for SNAP encapsulation

NOTE: If you set the MTU of a port to a value lower than the global MTU and from 576 – 1499, the port fragments the packets. However, if the port’s MTU is exactly 1500 and this is larger than the global MTU, the port drops the packets.

NOTE: You must save the configuration change and then reload the software to enable jumbo support.

To change the MTU for interface 1/5 to 1000, enter the following commands:

```
FastIron(config)#int e 1/5
FastIron(config-if-1/5)#ip mtu 1000
FastIron(config-if-1/5)#write memory
FastIron(config-if-1/5)#end
FastIron#reload
```

Syntax: [no] ip mtu <num>

The <num> parameter specifies the MTU. Ethernet II packets can hold IP packets from 576 – 1500 bytes long. If jumbo mode is enabled, Ethernet II packets can hold IP packets up to 9216 bytes long. Ethernet SNAP packets can hold IP packets from 576 – 1492 bytes long. If jumbo mode is enabled, SNAP packets can hold IP packets up to 9216 bytes long. The default MTU for Ethernet II packets is 1500. The default MTU for SNAP packets is 1492.

Path MTU Discovery (RFC 1191) Support

Except for the FGS devices, Foundry devices support the path MTU discovery method described in RFC 1191. When the Foundry device receives an IP packet that has its Don't Fragment (DF) bit set, and the packet size is greater than the MTU value of the outbound interface, then the Foundry device returns an ICMP Destination Unreachable message to the source of the packet, with the Code indicating "fragmentation needed and DF set". The ICMP Destination Unreachable message includes the MTU of the outbound interface. The source host can use this information to help determine the maximum MTU of a path to a destination.

RFC 1191 is supported on all interfaces.

NOTE: IP MTU is not supported for FGS and FLS devices.

Changing the Router ID

In most configurations, a Layer 3 Switch has multiple IP addresses, usually configured on different interfaces. As a result, a Layer 3 Switch's identity to other devices varies depending on the interface to which the other device is attached. Some routing protocols, including Open Shortest Path First (OSPF) and Border Gateway Protocol version 4 (BGP4), identify a Layer 3 Switch by just one of the IP addresses configured on the Layer 3 Switch, regardless of the interfaces that connect the Layer 3 Switches. This IP address is the router ID.

NOTE: Routing Information Protocol (RIP) does not use the router ID.

NOTE: If you change the router ID, all current BGP4 sessions are cleared.

By default, the router ID on a Foundry Layer 3 Switch is one of the following:

- If the router has loopback interfaces, the default router ID is the IP address configured on the lowest numbered loopback interface configured on the Layer 3 Switch. For example, if you configure loopback interfaces 1, 2, and 3 as follows, the default router ID is 9.9.9.9/24:
    - Loopback interface 1, 9.9.9.9/24
    - Loopback interface 2, 4.4.4.4/24
    - Loopback interface 3, 1.1.1.1/24
- If the device does not have any loopback interfaces, the default router ID is the lowest numbered IP interface configured on the device.

If you prefer, you can explicitly set the router ID to any valid IP address. The IP address cannot be in use on another device in the network.

NOTE: Foundry Layer 3 Switches use the same router ID for both OSPF and BGP4. If the router is already configured for OSPF, you may want to use the router ID that is already in use on the router rather than set a new one. To display the router ID, enter the **show ip** CLI command at any CLI level or select the IP->General links from the Configure tree in the Web management interface.

To change the router ID, enter a command such as the following:

```
FastIron(config)#ip router-id 209.157.22.26
```

Syntax: ip router-id <ip-addr>

The <ip-addr> can be any valid, unique IP address.

NOTE: You can specify an IP address used for an interface on the Foundry Layer 3 Switch, but do not specify an IP address in use by another device.

Specifying a Single Source Interface for Telnet, TACACS/TACACS+, or RADIUS Packets

When the Layer 3 Switch originates a Telnet, TACACS/TACACS+, or RADIUS packet, the source address of the packet is the lowest-numbered IP address on the interface that sends the packet. You can configure the Layer 3 Switch to always the lowest-numbered IP address on a specific interface as the source addresses for these types of packets. When you configure the Layer 3 Switch to use a single source interface for all Telnet, TACACS/TACACS+, or RADIUS packets, the Layer 3 Switch uses the same IP address as the source for all packets of the specified type, regardless of the port(s) that actually sends the packets.

Identifying a single source IP address for Telnet, TACACS/TACACS+, or RADIUS packets provides the following benefits:

- If your Telnet, TACACS/TACACS+, or RADIUS server is configured to accept packets only from specific IP addresses, you can use this feature to simplify configuration of the server by configuring the Foundry device to always send the packets from the same link or source address.
- If you specify a loopback interface as the single source for Telnet, TACACS/TACACS+, or RADIUS packets, servers can receive the packets regardless of the states of individual links. Thus, if a link to the server becomes unavailable but the client or server can be reached through another link, the client or server still receives the packets, and the packets still have the source IP address of the loopback interface.

The software contains separate CLI commands for specifying the source interface for Telnet, TACACS/TACACS+, or RADIUS packets. You can configure a source interface for one or more of these types of packets separately.

To specify an Ethernet or a loopback or virtual interface as the source for all TACACS/TACACS+ packets from the device, use the following CLI method. The software uses the lowest-numbered IP address configured on the port or interface as the source IP address for TACACS/TACACS+ packets originated by the device.

The following sections show the syntax for specifying a single source IP address for Telnet, TACACS/TACACS+, and RADIUS packets.

Telnet Packets

To specify the lowest-numbered IP address configured on a virtual interface as the device's source for all Telnet packets, enter commands such as the following:

```
FastIron(config)#int loopback 2
FastIron(config-lbif-2)#ip address 10.0.0.2/24
FastIron(config-lbif-2)#exit
FastIron(config)#ip telnet source-interface loopback 2
```

The commands in this example configure loopback interface 2, assign IP address 10.0.0.2/24 to the interface, then designate the interface as the source for all Telnet packets from the Layer 3 Switch.

Syntax: ip telnet source-interface ethernet [<slotnum>/]<portnum> | loopback <num> | ve <num>

The <num> parameter is a loopback interface or virtual interface number. If you specify an Ethernet port, the <portnum> is the port's number (including the slot number, if you are configuring a Chassis device).

The following commands configure an IP interface on an Ethernet port and designate the address port as the source for all Telnet packets from the Layer 3 Switch.

```
FastIron(config)#interface ethernet 1/4
FastIron(config-if-1/4)#ip address 209.157.22.110/24
FastIron(config-if-1/4)#exit
FastIron(config)#ip telnet source-interface ethernet 1/4
```

TACACS/TACACS+ Packets

To specify the lowest-numbered IP address configured on a virtual interface as the device's source for all TACACS/TACACS+ packets, enter commands such as the following:

```
FastIron(config)#int ve 1
FastIron(config-vif-1)#ip address 10.0.0.3/24
FastIron(config-vif-1)#exit
FastIron(config)#ip tacacs source-interface ve 1
```

The commands in this example configure virtual interface 1, assign IP address 10.0.0.3/24 to the interface, then designate the interface as the source for all TACACS/TACACS+ packets from the Layer 3 Switch.

Syntax: ip tacacs source-interface ethernet [<slotnum>/]<portnum> | loopback <num> | ve <num>

The <num> parameter is a loopback interface or virtual interface number. If you specify an Ethernet port, the <portnum> is the port's number (including the slot number, if you are configuring a Chassis device).

RADIUS Packets

To specify the lowest-numbered IP address configured on a virtual interface as the device's source for all RADIUS packets, enter commands such as the following:

```
FastIron(config)#int ve 1
FastIron(config-vif-1)#ip address 10.0.0.3/24
FastIron(config-vif-1)#exit
FastIron(config)#ip radius source-interface ve 1
```

The commands in this example configure virtual interface 1, assign IP address 10.0.0.3/24 to the interface, then designate the interface as the source for all RADIUS packets from the Layer 3 Switch.

Syntax: ip radius source-interface ethernet [<slotnum>/]<portnum> | loopback <num> | ve <num>

The <num> parameter is a loopback interface or virtual interface number. If you specify an Ethernet port, the <portnum> is the port's number (including the slot number, if you are configuring a Chassis device).

Configuring ARP Parameters

Address Resolution Protocol (ARP) is a standard IP protocol that enables an IP Layer 3 Switch to obtain the MAC address of another device's interface when the Layer 3 Switch knows the IP address of the interface. ARP is enabled by default and cannot be disabled.

NOTE: Foundry Layer 2 Switches also support ARP. The description in "How ARP Works" also applies to ARP on Foundry Layer 2 Switches. However, the configuration options described later in this section apply only to Layer 3 Switches, not to Layer 2 Switches.

How ARP Works

A Layer 3 Switch needs to know a destination's MAC address when forwarding traffic, because the Layer 3 Switch encapsulates the IP packet in a Layer 2 packet (MAC layer packet) and sends the Layer 2 packet to a MAC interface on a device directly attached to the Layer 3 Switch. The device can be the packet's final destination or the next-hop router toward the destination.

The Layer 3 Switch encapsulates IP packets in Layer 2 packets regardless of whether the ultimate destination is locally attached or is multiple router hops away. Since the Layer 3 Switch's IP route table and IP forwarding cache contain IP address information but not MAC address information, the Layer 3 Switch cannot forward IP packets based solely on the information in the route table or forwarding cache. The Layer 3 Switch needs to know the MAC address that corresponds with the IP address of either the packet's locally attached destination or the next-hop router that leads to the destination.

For example, to forward a packet whose destination is multiple router hops away, the Layer 3 Switch must send the packet to the next-hop router toward its destination, or to a default route or default network route if the IP route table does not contain a route to the packet's destination. In each case, the Layer 3 Switch must encapsulate the packet and address it to the MAC address of a locally attached device, the next-hop router toward the IP packet's destination.

To obtain the MAC address required for forwarding a datagram, the Layer 3 Switch does the following:

- First, the Layer 3 Switch looks in the ARP cache (not the static ARP table) for an entry that lists the MAC address for the IP address. The ARP cache maps IP addresses to MAC addresses. The cache also lists the port attached to the device and, if the entry is dynamic, the age of the entry. A dynamic ARP entry enters the cache when the Layer 3 Switch receives an ARP reply or receives an ARP request (which contains the sender's IP address and MAC address). A static entry enters the ARP cache from the static ARP table (which is a separate table) when the interface for the entry comes up.

  To ensure the accuracy of the ARP cache, each dynamic entry has its own age timer. The timer is reset to zero each time the Layer 3 Switch receives an ARP reply or ARP request containing the IP address and MAC address of the entry. If a dynamic entry reaches its maximum allowable age, the entry times out and the software removes the entry from the table. Static entries do not age out and can be removed only by you.

- If the ARP cache does not contain an entry for the destination IP address, the Layer 3 Switch broadcasts an ARP request out all its IP interfaces. The ARP request contains the IP address of the destination. If the device with the IP address is directly attached to the Layer 3 Switch, the device sends an ARP response containing its MAC address. The response is a unicast packet addressed directly to the Layer 3 Switch. The Layer 3 Switch places the information from the ARP response into the ARP cache.

  ARP requests contain the IP address and MAC address of the sender, so all devices that receive the request learn the MAC address and IP address of the sender and can update their own ARP caches accordingly.

  NOTE: The ARP request broadcast is a MAC broadcast, which means the broadcast goes only to devices that are directly attached to the Layer 3 Switch. A MAC broadcast is not routed to other networks. However, some routers, including Foundry Layer 3 Switches, can be configured to reply to ARP requests from one network on behalf of devices on another network. See “Enabling Proxy ARP” on page 29-28.

NOTE: If the router receives an ARP request packet that it is unable to deliver to the final destination because of the ARP timeout and no ARP response is received (the Layer 3 Switch knows of no route to the destination address), the router sends an ICMP Host Unreachable message to the source.

Rate Limiting ARP Packets

Platform Support:

- FESX/FSX/FWSX devices running software release 03.0.01 or later

You can limit the number of ARP packets the Foundry device accepts during each second. By default, the software does not limit the number of ARP packets the device can receive. Since the device sends ARP packets to the CPU for processing, if a device in a busy network receives a high number of ARP packets in a short period of time, some CPU processing might be deferred while the CPU processes the ARP packets.

To prevent the CPU from becoming flooded by ARP packets in a busy network, you can restrict the number of ARP packets the device will accept each second. When you configure an ARP rate limit, the device accepts up to the maximum number of packets you specify, but drops additional ARP packets received during the one-second interval. When a new one-second interval starts, the counter restarts at zero, so the device again accepts up to the maximum number of ARP packets you specified, but drops additional packets received within the interval.

To limit the number of ARP packets the device will accept each second, enter a command such as the following at the global CONFIG level of the CLI:

```
FastIron(config)#rate-limit-arp 100
```

This command configures the device to accept up to 100 ARP packets each second. If the device receives more than 100 ARP packets during a one-second interval, the device drops the additional ARP packets during the remainder of that one-second interval.

Syntax: [no] rate-limit-arp <num>

The <num> parameter specifies the number of ARP packets and can be from 0 – 100. If you specify 0, the device will not accept any ARP packets.

NOTE: If you want to change a previously configured the ARP rate limiting policy, you must remove the previously configured policy using the **no rate-limit-arp** <num> command before entering the new policy.

Changing the ARP Aging Period

When the Layer 3 Switch places an entry in the ARP cache, the Layer 3 Switch also starts an aging timer for the entry. The aging timer ensures that the ARP cache does not retain learned entries that are no longer valid. An entry can become invalid when the device with the MAC address of the entry is no longer on the network.

The ARP age affects dynamic (learned) entries only, not static entries. The default ARP age is ten minutes. On Layer 3 Switches, you can change the ARP age to a value from 0 – 240 minutes. You cannot change the ARP age on Layer 2 Switches. If you set the ARP age to zero, aging is disabled and entries do not age out.

To globally change the ARP aging parameter to 20 minutes, enter the following command:

```
FastIron(config)#ip arp-age 20
```

Syntax: ip arp-age <num>

The <num> parameter specifies the number of minutes and can be from 0 – 240. The default is 10. If you specify 0, aging is disabled.

To override the globally configured IP ARP age on an individual interface, enter a command such as the following at the interface configuration level:

```
FastIron(config-if-e1000-1/1)#ip arp-age 30
```

Syntax: [no] ip arp-age <num>

The <num> parameter specifies the number of minutes and can be from 0 – 240. The default is the globally configured value, which is 10 minutes by default. If you specify 0, aging is disabled.

Enabling Proxy ARP

Proxy ARP allows a Layer 3 Switch to answer ARP requests from devices on one network on behalf of devices in another network. Since ARP requests are MAC-layer broadcasts, they reach only the devices that are directly connected to the sender of the ARP request. Thus, ARP requests do not cross routers.

For example, if Proxy ARP is enabled on a Layer 3 Switch connected to two subnets, 10.10.10.0/24 and 20.20.20.0/24, the Layer 3 Switch can respond to an ARP request from 10.10.10.69 for the MAC address of the device with IP address 20.20.20.69. In standard ARP, a request from a device in the 10.10.10.0/24 subnet cannot reach a device in the 20.20.20.0 subnet if the subnets are on different network cables, and thus is not answered.

NOTE: An ARP request from one subnet can reach another subnet when both subnets are on the same physical segment (Ethernet cable), since MAC-layer broadcasts reach all the devices on the segment.

Proxy ARP is disabled by default on Foundry Layer 3 Switches. This feature is not supported on Foundry Layer 2 Switches.

In releases prior to FSX 04.0.00, you enable proxy ARP at the Global CONFIG level of the CLI.

Beginning in software release FSX 04.0.00, you can enable proxy ARP at the Interface level, as well as at the Global CONFIG level, of the CLI.

NOTE: Configuring proxy ARP at the Interface level overrides the global configuration.

Enabling Proxy ARP Globally

To enable IP proxy ARP on a global basis, enter the following command:

```
FastIron(config)#ip proxy-arp
```

To again disable IP proxy ARP on a global basis, enter the following command:

```
FastIron(config)#no ip proxy-arp
```

Syntax: [no] ip proxy-arp

Platform Support:

- Layer 3 devices only – BL3 and L3

Enabling IP ARP on an Interface

Platform Support:

- FESX/FSX/FWSX devices running software release 04.0.00 and later

NOTE: Configuring proxy ARP at the Interface level overrides the global configuration.

To enable IP proxy ARP on an interface, enter the following command:

```
FastIron(config)#int e 5
FastIron(config-if-e1000-5)#ip proxy-arp enable
```

To again disable IP proxy ARP on an interface, enter the following command:

```
FastIron(config)#int e 5
FastIron(config-if-e1000-5)#ip proxy-arp disable
```

Syntax: [no] ip proxy-arp enable | disable

Platform Support:

- FESX/FSX/FWSX devices running software release 04.0.00 and later – BL3 and L3

Enabling Local Proxy ARP

NOTE: This feature is not supported in the FGS.

Foundry devices support Proxy Address Resolution Protocol (***Proxy ARP***), a feature that enables router ports to respond to ARP requests for subnets it can reach. However, router ports will not respond to ARP requests for IP addresses in the same subnet as the incoming ports. Software release 02.3.03 resolves this issue with the introduction of Local Proxy ARP per IP interface. ***Local Proxy ARP*** enables router ports to reply to ARP requests for IP addresses within the same subnet and to forward all traffic between hosts in the subnet.

When Local Proxy ARP is enabled on a router port, the port will respond to ARP requests for IP addresses within the same subnet, if it has ARP entries for the destination IP addresses in the ARP cache. If it does not have ARP entries for the IP addresses, the port will attempt to resolve them by broadcasting its own ARP requests.

Local Proxy ARP is disabled by default. To use Local Proxy ARP, Proxy ARP (CLI command **ip proxy-arp**) must be enabled globally on the Foundry device. You can enter the CLI command to enable Local Proxy ARP even though Proxy ARP is not enabled, however, the configuration will not take effect until you enable Proxy ARP.

Use the **show run** command to view the ports on which Local Proxy ARP is enabled.

To enable Local Proxy ARP, enter commands such as the following:

```
FastIron(config)#int e 4
FastIron(config-if-e1000-4)#ip local-proxy-arp
```

Syntax: [no] ip local-proxy-arp

Use the **no** form of the command to disable Local Proxy ARP.

Creating Static ARP Entries

Foundry Layer 3 Switches have a static ARP table, in addition to the regular ARP cache. The static ARP table contains entries that you configure.

Static entries are useful in cases where you want to pre-configure an entry for a device that is not connected to the Layer 3 Switch, or you want to prevent a particular entry from aging out. The software removes a dynamic entry from the ARP cache if the ARP aging interval expires before the entry is refreshed. Static entries do not age out, regardless of whether the Foundry device receives an ARP request from the device that has the entry's address.

NOTE: You cannot create static ARP entries on a Layer 2 Switch.

The maximum number of static ARP entries you can configure depends on the product. See "Changing the Maximum Number of Entries the Static ARP Table Can Hold" on page 29-30.

To display the ARP cache and static ARP table, see the following:

- To display the ARP table, see "Displaying the ARP Cache" on page 29-70.

- To display the static ARP table, see “Displaying the Static ARP Table” on page 29-71.

To create a static ARP entry, enter a command such as the following:

```
FastIron(config)#arp 1 192.53.4.2 1245.7654.2348 e 1/2
```

Syntax: arp <num> <ip-addr> <mac-addr> ethernet [<slotnum>/]<portnum>

The <num> parameter specifies the entry number. You can specify a number from 1 up to the maximum number of static entries allowed on the device.

The <ip-addr> command specifies the IP address of the device that has the MAC address of the entry.

The <mac-addr> parameter specifies the MAC address of the entry.

The <slotnum> parameter is required on chassis devices.

The <portnum> command specifies the port number attached to the device that has the MAC address of the entry.

Changing the Maximum Number of Entries the Static ARP Table Can Hold

Table 29.5 on page 29-30 lists the default maximum and configurable maximum number of entries in the static ARP table that are supported on each type of Foundry Layer 3 Switch. If you need to change the maximum number of entries supported on a Layer 3 Switch, use either of the following methods.

NOTE: You must save the configuration to the startup-config file and reload the software after changing the static ARP table size to place the change into effect.

NOTE: The basic procedure for changing the static ARP table size is the same as the procedure for changing other configurable cache or table sizes. See the section “Displaying and Modifying System Parameter Default Settings” on page 8-11.

To increase the maximum number of static ARP table entries you can configure on a Foundry Layer 3 Switch, enter commands such as the following at the global CONFIG level of the CLI:

```
FastIron(config)#system-max ip-static-arp 1000
FastIron(config)#write memory
FastIron(config)#end
FastIron#reload
```

Syntax: system-max ip-static-arp <num>

The <num> parameter indicates the maximum number of static ARP entries and can be a number in one of the ranges shown in Table 29.5, depending on the device you are configuring. Table 29.5 lists the default maximum and range of configurable maximums for static ARP table entries supported on a Foundry Layer 3 Switch.

Table 29.5: Static ARP Entry Support

Default Maximum	Configurable Minimum	Configurable Maximum
512	512	1024

Configuring Forwarding Parameters

The following configurable parameters control the forwarding behavior of Foundry Layer 3 Switches:

- Time-To-Live (TTL) threshold
- Forwarding of directed broadcasts
- Forwarding of source-routed packets
- Ones-based and zero-based broadcasts

All these parameters are global and thus affect all IP interfaces configured on the Layer 3 Switch.

To configure these parameters, use the procedures in the following sections.

Changing the TTL Threshold

The TTL threshold prevents routing loops by specifying the maximum number of router hops an IP packet originated by the Layer 3 Switch can travel through. Each device capable of forwarding IP that receives the packet decrements (decreases) the packet's TTL by one. If a device receives a packet with a TTL of 1 and reduces the TTL to zero, the device drops the packet.

The default TTL is 64. You can change the TTL to a value from 1– 255.

To modify the TTL threshold to 25, enter the following commands:

```
FastIron(config)#ip ttl 25
```

Syntax: ip ttl <1-255>

Enabling Forwarding of Directed Broadcasts

A directed broadcast is an IP broadcast to all devices within a single directly-attached network or subnet. A net-directed broadcast goes to all devices on a given network. A subnet-directed broadcast goes to all devices within a given subnet.

NOTE: A less common type, the all-subnets broadcast, goes to all directly-attached subnets. Forwarding for this broadcast type also is supported, but most networks use IP multicasting instead of all-subnet broadcasting.

Forwarding for all types of IP directed broadcasts is disabled by default. You can enable forwarding for all types if needed. You cannot enable forwarding for specific broadcast types.

To enable forwarding of IP directed broadcasts, enter the following command:

```
FastIron(config)#ip directed-broadcast
```

Syntax: [no] ip directed-broadcast

Foundry software makes the forwarding decision based on the router's knowledge of the destination network prefix. Routers cannot determine that a message is unicast or directed broadcast apart from the destination network prefix. The decision to forward or not forward the message is by definition only possible in the last hop router.

To disable the directed broadcasts, enter the following command in the CONFIG mode:

```
FastIron(config)#no ip directed-broadcast
```

To enable directed broadcasts on an individual interface instead of globally for all interfaces, enter commands such as the following:

```
FastIron(config)#interface ethernet 1/1
FastIron(config-if-1/1)#ip directed-broadcast
```

Syntax: [no] ip directed-broadcast

Disabling Forwarding of IP Source-Routed Packets

A source-routed packet specifies the exact router path for the packet. The packet specifies the path by listing the IP addresses of the router interfaces through which the packet must pass on its way to the destination. The Layer 3 Switch supports both types of IP source routing:

- Strict source routing – requires the packet to pass through only the listed routers. If the Layer 3 Switch receives a strict source-routed packet but cannot reach the next hop interface specified by the packet, the Layer 3 Switch discards the packet and sends an ICMP Source-Route-Failure message to the sender.

  NOTE: The Layer 3 Switch allows you to disable sending of the Source-Route-Failure messages. See "Disabling ICMP Messages" on page 29-32.

- Loose source routing – requires that the packet pass through all of the listed routers but also allows the packet to travel through other routers, which are not listed in the packet.

The Layer 3 Switch forwards both types of source-routed packets by default. To disable the feature, use either of the following methods. You cannot enable or disable strict or loose source routing separately.

To disable forwarding of IP source-routed packets, enter the following command:

```
FastIron(config)#no ip source-route
```

Syntax: [no] ip source-route

To re-enable forwarding of source-routed packets, enter the following command:

```
FastIron(config)#ip source-route
```

Enabling Support for Zero-Based IP Subnet Broadcasts

By default, the Layer 3 Switch treats IP packets with all ones in the host portion of the address as IP broadcast packets. For example, the Layer 3 Switch treats IP packets with 209.157.22.255/24 as the destination IP address as IP broadcast packets and forwards the packets to all IP hosts within the 209.157.22.x subnet (except the host that sent the broadcast packet to the Layer 3 Switch).

Most IP hosts are configured to receive IP subnet broadcast packets with all ones in the host portion of the address. However, some older IP hosts instead expect IP subnet broadcast packets that have all zeros instead of all ones in the host portion of the address. To accommodate this type of host, you can enable the Layer 3 Switch to treat IP packets with all zeros in the host portion of the destination IP address as broadcast packets.

NOTE: When you enable the Layer 3 Switch for zero-based subnet broadcasts, the Layer 3 Switch still treats IP packets with all ones the host portion as IP subnet broadcasts too. Thus, the Layer 3 Switch can be configured to support all ones only (the default) or all ones ***and*** all zeroes.

NOTE: This feature applies only to IP subnet broadcasts, not to local network broadcasts. The local network broadcast address is still expected to be all ones.

To enable the Layer 3 Switch for zero-based IP subnet broadcasts in addition to ones-based IP subnet broadcasts, enter the following command.

```
FastIron(config)#ip broadcast-zero
FastIron(config)#write memory
FastIron(config)#end
FastIron#reload
```

NOTE: You must save the configuration and reload the software to place this configuration change into effect.

Syntax: [no] ip broadcast-zero

Disabling ICMP Messages

Foundry devices are enabled to reply to ICMP echo messages and send ICMP Destination Unreachable messages by default.

You can selectively disable the following types of Internet Control Message Protocol (ICMP) messages:

- Echo messages (ping messages) – The Layer 3 Switch replies to IP pings from other IP devices.
- Destination Unreachable messages – If the Layer 3 Switch receives an IP packet that it cannot deliver to its destination, the Layer 3 Switch discards the packet and sends a message back to the device that sent the packet to the Layer 3 Switch. The message informs the device that the destination cannot be reached by the Layer 3 Switch.

Disabling Replies to Broadcast Ping Requests

By default, Foundry devices are enabled to respond to broadcast ICMP echo packets, which are ping requests.

To disable response to broadcast ICMP echo packets (ping requests), enter the following command:

```
FastIron(config)#no ip icmp echo broadcast-request
```

Syntax: [no] ip icmp echo broadcast-request

If you need to re-enable response to ping requests, enter the following command:

```
FastIron(config)#ip icmp echo broadcast-request
```

Disabling ICMP Destination Unreachable Messages

By default, when a Foundry device receives an IP packet that the device cannot deliver, the device sends an ICMP Unreachable message back to the host that sent the packet. You can selectively disable a Foundry device's response to the following types of ICMP Unreachable messages:

- Administration – The packet was dropped by the Foundry device due to a filter or ACL configured on the device.
- Fragmentation-needed – The packet has the Don't Fragment bit set in the IP Flag field, but the Foundry device cannot forward the packet without fragmenting it.
- Host – The destination network or subnet of the packet is directly connected to the Foundry device, but the host specified in the destination IP address of the packet is not on the network.
- Port – The destination host does not have the destination TCP or UDP port specified in the packet. In this case, the host sends the ICMP Port Unreachable message to the Foundry device, which in turn sends the message to the host that sent the packet.
- Protocol – The TCP or UDP protocol on the destination host is not running. This message is different from the Port Unreachable message, which indicates that the protocol is running on the host but the requested protocol port is unavailable.
- Source-route-failure – The device received a source-routed packet but cannot locate the next-hop IP address indicated in the packet's Source-Route option.

You can disable the Foundry device from sending these types of ICMP messages on an individual basis. To do so, use the following CLI method.

NOTE: Disabling an ICMP Unreachable message type does not change the Foundry device's ability to forward packets. Disabling ICMP Unreachable messages prevents the device from generating or forwarding the Unreachable messages.

To disable all ICMP Unreachable messages, enter the following command:

```
FastIron(config)#no ip icmp unreachable
```

Syntax: [no] ip icmp unreachable [host | protocol | administration | fragmentation-needed | port | source-route-fail]

- If you enter the command without specifying a message type (as in the example above), all types of ICMP Unreachable messages listed above are disabled. If you want to disable only specific types of ICMP Unreachable messages, you can specify the message type. To disable more than one type of ICMP message, enter the **no ip icmp unreachable** command for each messages type.
- The **administration** parameter disables ICMP Unreachable (caused by Administration action) messages.
- The **fragmentation-needed** parameter disables ICMP Fragmentation-Needed But Don't-Fragment Bit Set messages.
- The **host** parameter disables ICMP Host Unreachable messages.
- The **port** parameter disables ICMP Port Unreachable messages.
- The **protocol** parameter disables ICMP Protocol Unreachable messages.
- The **source-route-fail** parameter disables ICMP Unreachable (caused by Source-Route-Failure) messages.

To disable ICMP Host Unreachable messages but leave the other types of ICMP Unreachable messages enabled, enter the following commands instead of the command shown above:

```
FastIron(config)#no ip icmp unreachable host
```

If you have disabled all ICMP Unreachable message types but you want to re-enable certain types, for example ICMP Host Unreachable messages, you can do so by entering the following command:

```
FastIron(config)#ip icmp unreachable host
```

Configuring Static Routes

The IP route table can receive routes from the following sources:

- Directly-connected networks – When you add an IP interface, the Layer 3 Switch automatically creates a route for the network the interface is in.
- RIP – If RIP is enabled, the Layer 3 Switch can learn about routes from the advertisements other RIP routers send to the Layer 3 Switch. If the route has a lower administrative distance than any other routes from different sources to the same destination, the Layer 3 Switch places the route in the IP route table.
- OSPF – See RIP, but substitute "OSPF" for "RIP".
- BGP4 – See RIP, but substitute "BGP4" for "RIP".
- Default network route – A statically configured default route that the Layer 3 Switch uses if other default routes to the destination are not available. See "Configuring a Default Network Route" on page 29-41.
- Statically configured route – You can add routes directly to the route table. When you add a route to the IP route table, you are creating a static IP route. This section describes how to add static routes to the IP route table.

Static Route Types

You can configure the following types of static IP routes:

- Standard – the static route consists of the destination network address and network mask, and the IP address of the next-hop gateway. You can configure multiple standard static routes with the same metric for load sharing or with different metrics to provide a primary route and backup routes.
- Interface-based – the static route consists of the destination network address and network mask, and the Layer 3 Switch interface through which you want the Layer 3 Switch to send traffic for the route. Typically, this type of static route is for directly attached destination networks.
- Null – the static route consists of the destination network address and network mask, and the "null0" parameter. Typically, the null route is configured as a backup route for discarding traffic if the primary route is unavailable.

Static IP Route Parameters

When you configure a static IP route, you must specify the following parameters:

- The IP address and network mask for the route's destination network.
- The route's path, which can be one of the following:
  - The IP address of a next-hop gateway
  - An Ethernet port
  - A virtual interface (a routing interface used by VLANs for routing Layer 3 protocol traffic among one another)
  - A "null" interface. The Layer 3 Switch drops traffic forwarded to the null interface.

You also can specify the following optional parameters:

- The route's metric – The value the Layer 3 Switch uses when comparing this route to other routes in the IP route table to the same destination. The metric applies only to routes that the Layer 3 Switch has already placed in the IP route table. The default metric for static IP routes is 1.
- The route's administrative distance – The value that the Layer 3 Switch uses to compare this route with routes from other route sources to the same destination before placing a route in the IP route table. This parameter does not apply to routes that are already in the IP route table. The default administrative distance for static IP routes is 1.

The default metric and administrative distance values ensure that the Layer 3 Switch always prefers static IP routes over routes from other sources to the same destination.

Multiple Static Routes to the Same Destination Provide Load Sharing and Redundancy

You can add multiple static routes for the same destination network to provide one or more of the following benefits:

- IP load balancing – When you add multiple IP static routes for the same destination to different next-hop gateways, and the routes each have the same metric and administrative distance, the Layer 3 Switch can load balance traffic to the routes' destination. For information about IP load balancing, see "Configuring IP Load Sharing" on page 29-42.
- Path redundancy – When you add multiple static IP routes for the same destination, but give the routes different metrics or administrative distances, the Layer 3 Switch uses the route with the lowest administrative distance by default, but uses another route to the same destination if the first route becomes unavailable.

See the following sections for examples and configuration information:

- "Configuring Load Balancing and Redundancy Using Multiple Static Routes to the Same Destination" on page 29-38
- "Configuring Standard Static IP Routes and Interface or Null Static Routes to the Same Destination" on page 29-39

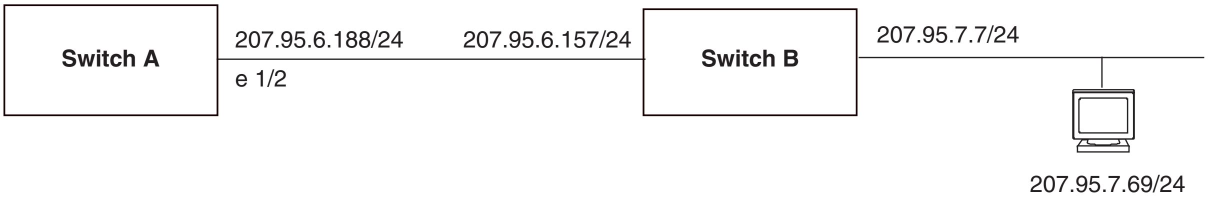
Static Route States Follow Port States

IP static routes remain in the IP route table only so long as the port or virtual interface used by the route is available. If the port or virtual routing interface becomes unavailable, the software removes the static route from the IP route table. If the port or virtual routing interface becomes available again later, the software adds the route back to the route table.

This feature allows the Layer 3 Switch to adjust to changes in network topology. The Layer 3 Switch does not continue trying to use routes on unavailable paths but instead uses routes only when their paths are available.

Figure 29.3 shows an example of a network containing a static route. The static route is configured on Switch A, as shown in the CLI example following the figure.

Figure 29.3 Example of a Static Route



The following command configures a static route to 207.95.7.0, using 207.95.6.157 as the next-hop gateway.

```
FastIron(config)#ip route 207.95.7.0/24 207.95.6.157
```

When you configure a static IP route, you specify the destination address for the route and the next-hop gateway or Layer 3 Switch interface through which the Layer 3 Switch can reach the route. The Layer 3 Switch adds the route to the IP route table. In this case, Switch A knows that 207.95.6.157 is reachable through port 1/2, and also assumes that local interfaces within that subnet are on the same port. Switch A deduces that IP interface 207.95.7.188 is also on port 1/2.

The software automatically removes a static IP route from the IP route table if the port used by that route becomes unavailable. When the port becomes available again, the software automatically re-adds the route to the IP route table.

Configuring a Static IP Route

To configure an IP static route with a destination address of 192.0.0.0 255.0.0.0 and a next-hop router IP address of 195.1.1.1, enter the following commands:

```
FastIron(config)#ip route 192.0.0.0 255.0.0.0 195.1.1.1
```

To configure a static IP route with an Ethernet port instead of a next-hop address, enter a command such as the following.

```
FastIron(config)#ip route 192.128.2.69 255.255.255.0 ethernet 4/1
```

The command in the example above configures a static IP route for destination network 192.128.2.69/24. Since an Ethernet port is specified instead of a gateway IP address as the next hop, the Layer 3 Switch always forwards traffic for the 192.128.2.69/24 network to port 4/1. The command in the following example configures an IP static route that uses virtual interface 3 as its next hop.

```
FastIron(config)#ip route 192.128.2.71 255.255.255.0 ve 3
```

The command in the following example configures an IP static route that uses port 2/2 as its next hop.

```
FastIron(config)#ip route 192.128.2.73 255.255.255.0 ethernet 2/2
```

Syntax: ip route <dest-ip-addr> <dest-mask>
<next-hop-ip-addr> |
ethernet [<slotnum>/]<portnum> | ve <num>
[<metric>] [distance <num>]

or

Syntax: ip route <dest-ip-addr>/<mask-bits>
<next-hop-ip-addr> |
ethernet [<slotnum>/]<portnum> | ve <num>
[<metric>] [distance <num>]

The <dest-ip-addr> is the route's destination. The <dest-mask> is the network mask for the route's destination IP address. Alternatively, you can specify the network mask information by entering a forward slash followed by the number of bits in the network mask. For example, you can enter 192.0.0.0 255.255.255.0 as 192.0.0.0/.24.

The <next-hop-ip-addr> is the IP address of the next-hop router (gateway) for the route.

If you do not want to specify a next-hop IP address, you can instead specify a port or interface number on the Layer 3 Switch. The <num> parameter is a virtual interface number. If you instead specify an Ethernet port, the <portnum> is the port's number (including the slot number, if you are configuring a Chassis device). In this case, the Layer 3 Switch forwards packets destined for the static route's destination network to the specified interface. Conceptually, this feature makes the destination network like a directly connected network, associated with a specific Layer 3 Switch interface.

NOTE: The port or virtual interface you use for the static route's next hop must have at least one IP address configured on it. The address does not need to be in the same subnet as the destination network.

The <metric> parameter can be a number from 1 – 16. The default is 1.

NOTE: If you specify 16, RIP considers the metric to be infinite and thus also considers the route to be unreachable.

The **distance** <num> parameter specifies the administrative distance of the route. When comparing otherwise equal routes to a destination, the Layer 3 Switch prefers lower administrative distances over higher ones, so make sure you use a low value for your default route. The default is 1.

NOTE: The Layer 3 Switch will replace the static route if the it receives a route with a lower administrative distance. See "Changing Administrative Distances" on page 38-28 for a list of the default administrative distances for all types of routes.

NOTE: You can also assign the default router as the destination by entering 0.0.0.0 0.0.0.0 xxx.xxx.xxx.xxx.

Configuring a "Null" Route

You can configure the Layer 3 Switch to drop IP packets to a specific network or host address by configuring a "null" (sometimes called "null0") static route for the address. When the Layer 3 Switch receives a packet destined for the address, the Layer 3 Switch drops the packet instead of forwarding it.

To configure a null static route, use the following CLI method.

To configure a null static route to drop packets destined for network 209.157.22.x, enter the following commands.

```
FastIron(config)#ip route 209.157.22.0 255.255.255.0 null0
FastIron(config)#write memory
```

Syntax: ip route <ip-addr> <ip-mask> null0 [<metric>] [distance <num>]

or

Syntax: ip route <ip-addr>/<mask-bits> null0 [<metric>] [distance <num>]

To display the maximum value for your device, enter the **show default values** command. The maximum number of static IP routes the system can hold is listed in the ip-static-route row in the System Parameters section of the display. To change the maximum value, use the **system-max ip-static-route** <num> command at the global CONFIG level.

The <ip-addr> parameter specifies the network or host address. The Layer 3 Switch will drop packets that contain this address in the destination field instead of forwarding them.

The <ip-mask> parameter specifies the network mask. Ones are significant bits and zeros allow any value. For example, the mask 255.255.255.0 matches on all hosts within the Class C subnet address specified by <ip-addr>. Alternatively, you can specify the number of bits in the network mask. For example, you can enter 209.157.22.0/24 instead of 209.157.22.0 255.255.255.0.

The **null0** parameter indicates that this is a null route. You must specify this parameter to make this a null route.

The <metric> parameter adds a cost to the route. You can specify from 1 – 16. The default is 1.

The distance <num> parameter configures the administrative distance for the route. You can specify a value from 1 – 255. The default is 1. The value 255 makes the route unusable.

NOTE: The last two parameters are optional and do not affect the null route, unless you configure the administrative distance to be 255. In this case, the route is not used and the traffic might be forwarded instead of dropped.

Configuring Load Balancing and Redundancy Using Multiple Static Routes to the Same Destination

You can configure multiple static IP routes to the same destination, for the following benefits:

- **IP load sharing** – If you configure more than one static route to the same destination, and the routes have different next-hop gateways but have the same metrics, the Layer 3 Switch load balances among the routes using basic round-robin. For example, if you configure two static routes with the same metrics but to different gateways, the Layer 3 Switch alternates between the two routes. For information about IP load balancing, see "Configuring IP Load Sharing" on page 29-42.
- **Backup Routes** – If you configure multiple static IP routes to the same destination, but give the routes different next-hop gateways and different metrics, the Layer 3 Switch will always use the route with the lowest metric. If this route becomes unavailable, the Layer 3 Switch will fail over to the static route with the next-lowest metric, and so on.

NOTE: You also can bias the Layer 3 Switch to select one of the routes by configuring them with different administrative distances. However, make sure you do not give a static route a higher administrative distance than other types of routes, unless you want those other types to be preferred over the static route. For a list of the default administrative distances, see "Changing Administrative Distances" on page 38-28.

The steps for configuring the static routes are the same as described in the previous section. The following sections provide examples.

To configure multiple static IP routes, enter commands such as the following.

```
FastIron(config)#ip route 192.128.2.69 255.255.255.0 209.157.22.1
FastIron(config)#ip route 192.128.2.69 255.255.255.0 192.111.10.1
```

The commands in the example above configure two static IP routes. The routes go to different next-hop gateways but have the same metrics. These commands use the default metric value (1), so the metric is not specified. These static routes are used for load sharing among the next-hop gateways.

The following commands configure static IP routes to the same destination, but with different metrics. The route with the lowest metric is used by default. The other routes are backups in case the first route becomes unavailable. The Layer 3 Switch uses the route with the lowest metric if the route is available.

```
FastIron(config)#ip route 192.128.2.69 255.255.255.0 209.157.22.1
FastIron(config)#ip route 192.128.2.69 255.255.255.0 192.111.10.1 2
FastIron(config)#ip route 192.128.2.69 255.255.255.0 201.1.1.1 3
```

In this example, each static route has a different metric. The metric is not specified for the first route, so the default (1) is used. A metric is specified for the second and third static IP routes. The second route has a metric of two and the third route has a metric of 3. Thus, the second route is used only of the first route (which has a metric of 1) becomes unavailable. Likewise, the third route is used only if the first and second routes (which have lower metrics) are both unavailable.

For complete syntax information, see “Configuring a Static IP Route” on page 29-36.

Configuring Standard Static IP Routes and Interface or Null Static Routes to the Same Destination

You can configure a null0 or interface-based static route to a destination and also configure a normal static route to the same destination, so long as the route metrics are different.

When the Layer 3 Switch has multiple routes to the same destination, the Layer 3 Switch always prefers the route with the lowest metric. Generally, when you configure a static route to a destination network, you assign the route a low metric so that the Layer 3 Switch prefers the static route over other routes to the destination.

This feature is especially useful for the following configurations. These are not the only allowed configurations but they are typical uses of this enhancement.

- When you want to ensure that if a given destination network is unavailable, the Layer 3 Switch drops (forwards to the null interface) traffic for that network instead of using alternate paths to route the traffic. In this case, assign the normal static route to the destination network a lower metric than the null route.
- When you want to use a specific interface by default to route traffic to a given destination network, but want to allow the Layer 3 Switch to use other interfaces to reach the destination network if the path that uses the default interface becomes unavailable. In this case, give the interface route a lower metric than the normal static route.

NOTE: You cannot add a null or interface-based static route to a network if there is already a static route of any type with the same metric you specify for the null or interface-based route.

Figure 29.4 shows an example of two static routes configured for the same destination network. In this example, one of the routes is a standard static route and has a metric of 1. The other static route is a null route and has a higher metric than the standard static route. The Layer 3 Switch always prefers the static route with the lower metric. In this example, the Layer 3 Switch always uses the standard static route for traffic to destination network 192.168.7.0/24, unless that route becomes unavailable, in which case the Layer 3 Switch sends traffic to the null route instead.

Figure 29.4 Standard and Null Static Routes to the Same Destination Network

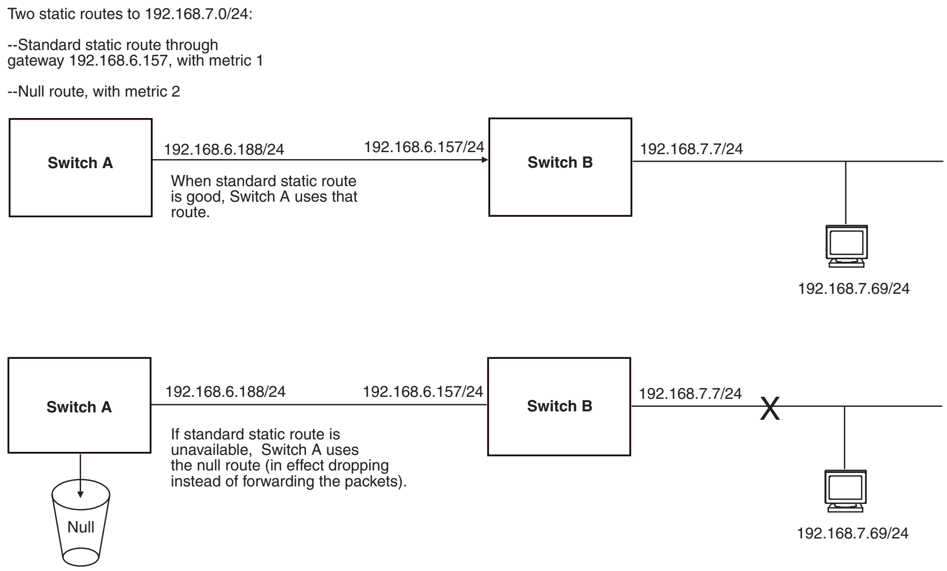
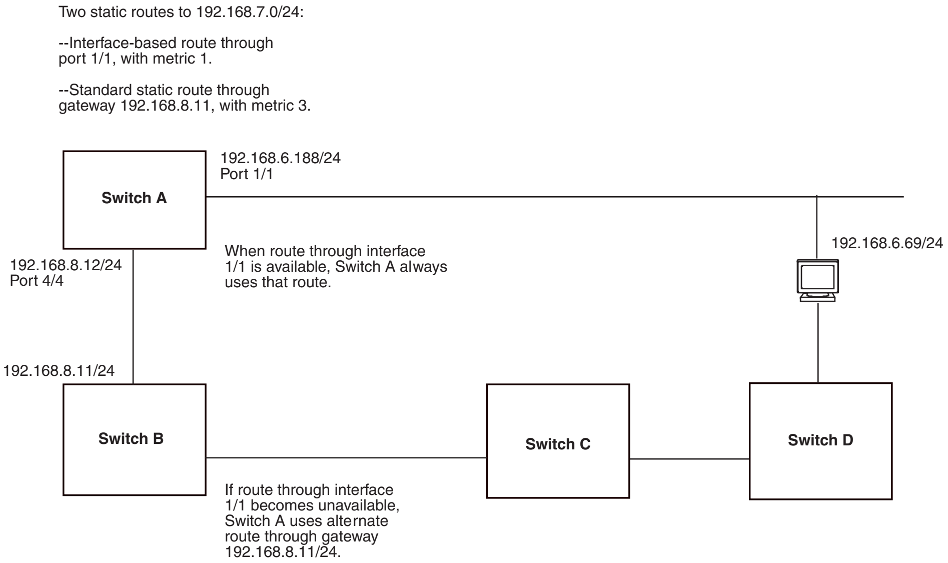

Two static routes to 192.168.7.0/24:
--Standard static route through gateway 192.168.6.157, with metric 1
--Null route, with metric 2
Switch A
192.168.6.188/24
192.168.6.157/24
Switch B
192.168.7.7/24
When standard static route is good, Switch A uses that route.
192.168.7.69/24
Switch A
192.168.6.188/24
192.168.6.157/24
Switch B
192.168.7.7/24
X
If standard static route is unavailable, Switch A uses the null route (in effect dropping instead of forwarding the packets).
Null
192.168.7.69/24

Figure 29.5 shows another example of two static routes. In this example, a standard static route and an interface-based static route are configured for destination network 192.168.6.0/24. The interface-based static route has a lower metric than the standard static route. As a result, the Layer 3 Switch always prefers the interface-based route when the route is available. However, if the interface-based route becomes unavailable, the Layer 3 Switch still forwards the traffic toward the destination using an alternate route through gateway 192.168.8.11/24.

Figure 29.5 Standard and Interface Routes to the Same Destination Network


Two static routes to 192.168.7.0/24:
--Interface-based route through port 1/1, with metric 1.
--Standard static route through gateway 192.168.8.11, with metric 3.
Switch A
192.168.6.188/24
Port 1/1
192.168.6.69/24
When route through interface 1/1 is available, Switch A always uses that route.
192.168.8.12/24
Port 4/4
192.168.8.11/24
Switch B
Switch C
Switch D
If route through interface 1/1 becomes unavailable, Switch A uses alternate route through gateway 192.168.8.11/24.

To configure a standard static IP route and a null route to the same network as shown in Figure 29.4 on page 29-40, enter commands such as the following:

```
FastIron(config)#ip route 192.168.7.0/24 192.168.6.157/24 1
FastIron(config)#ip route 192.168.7.0/24 null0 3
```

The first command configures a standard static route, which includes specification of the next-hop gateway. The command also gives the standard static route a metric of 1, which causes the Layer 3 Switch to always prefer this route when the route is available.

The second command configures another static route for the same destination network, but the second route is a null route. The metric for the null route is 3, which is higher than the metric for the standard static route. If the standard static route is unavailable, the software uses the null route.

For complete syntax information, see “Configuring a Static IP Route” on page 29-36.

To configure a standard static route and an interface-based route to the same destination, enter commands such as the following:

```
FastIron(config)#ip route 192.168.6.0/24 ethernet 1/1 1
FastIron(config)#ip route 192.168.6.0/24 192.168.8.11/24 3
```

The first command configured an interface-based static route through Ethernet port 1/1. The command assigns a metric of 1 to this route, causing the Layer 3 Switch to always prefer this route when it is available. If the route becomes unavailable, the Layer 3 Switch uses an alternate route through the next-hop gateway 192.168.8.11/24.

Configuring a Default Network Route

The Layer 3 Switch enables you to specify a candidate default route without the need to specify the next hop gateway. If the IP route table does not contain an explicit default route (for example, 0.0.0.0/0) or propagate an explicit default route through routing protocols, the software can use the default network route as a default route instead.

When the software uses the default network route, it also uses the default network route's next hop gateway as the gateway of last resort.

This feature is especially useful in environments where network topology changes can make the next hop gateway unreachable. This feature allows the Layer 3 Switch to perform default routing even if the default network route's default gateway changes.

The feature thus differs from standard default routes. When you configure a standard default route, you also specify the next hop gateway. If a topology change makes the gateway unreachable, the default route becomes unusable.

For example, if you configure 10.10.10.0/24 as a candidate default network route, if the IP route table does not contain an explicit default route (0.0.0.0/0), the software uses the default network route and automatically uses that route's next hop gateway as the default gateway. If a topology change occurs and as a result the default network route's next hop gateway changes, the software can still use the default network route. To configure a default network route, use the following CLI method.

If you configure more than one default network route, the Layer 3 Switch uses the following algorithm to select one of the routes:

1. Use the route with the lowest administrative distance.
2. If the administrative distances are equal:
   - Are the routes from different routing protocols (RIP, OSPF, or BGP4)? If so, use the route with the lowest IP address.
   - If the routes are from the same routing protocol, use the route with the best metric. The meaning of “best” metric depends on the routing protocol:
   - RIP – The metric is the number of hops (additional routers) to the destination. The best route is the route with the fewest hops.
   - OSPF – The metric is the path cost associated with the route. The path cost does not indicate the number of hops but is instead a numeric value associated with each route. The best route is the route with the lowest path cost.
   - BGP4 – The metric is the Multi-exit Discriminator (MED) associated with the route. The MED applies to routes that have multiple paths through the same AS. The best route is the route with the lowest MED.

Configuring a Default Network Route

You can configure up to four default network routes.

To configure a default network route, enter commands such as the following:

```
FastIron(config)#ip default-network 209.157.22.0
FastIron(config)#write memory
```

Syntax: ip default-network <ip-addr>

The <ip-addr> parameter specifies the network address.

To verify that the route is in the route table, enter the following command at any level of the CLI:

```
FastIron#show ip route
Total number of IP routes: 2
Start index: 1  B:BGP D:Connected  R:RIP  S:Static  O:OSPF *:Candidate default
      Destination       NetMask           Gateway         Port   Cost   Type
1     209.157.20.0      255.255.255.0     0.0.0.0         lb1    1      D
2     209.157.22.0      255.255.255.0     0.0.0.0         4/11   1      *D
```

This example shows two routes. Both of the routes are directly attached, as indicated in the Type column. However, one of the routes is shown as type "*D", with an asterisk (*). The asterisk indicates that this route is a candidate default network route.

Configuring IP Load Sharing

The IP route table can contain more than one path to a given destination. When this occurs, the Layer 3 Switch selects the path with the lowest cost as the path for forwarding traffic to the destination. If the IP route table contains more than one path to a destination and the paths each have the lowest cost, then the Layer 3 Switch uses ***IP load sharing*** to select a path to the destination.[1]

On X Series devices, IP load sharing uses a hashing algorithm based on the source IP address, destination IP address, and protocol field in the IP header.

In software releases 02.1.00 and later, you can enable a Layer 3 Switch to load balance across up to six equal-cost paths. In software releases prior to 02.1.00, you can enable a Layer 3 Switch to load balance across up to eight equal-cost paths. The default maximum number of equal-cost load sharing paths is four.

NOTE: IP load sharing is based on next-hop routing, and not on source routing.

NOTE: The term "path" refers to the next-hop router to a destination, not to the entire route to a destination. Thus, when the software compares multiple equal-cost paths, the software is comparing paths that use different next-hop routers, with equal costs, to the same destination.

In many contexts, the terms "route" and "path" mean the same thing. Most of the user documentation uses the term "route" throughout. The term "path" is used in this section to refer to an individual next-hop router to a destination, while the term "route" refers collectively to the multiple paths to the destination. Load sharing applies when the IP route table contains multiple, equal-cost paths to a destination.

NOTE: Foundry devices also perform load sharing among the ports in aggregate links. See "Trunk Group Load Sharing" on page 13-7.

How Multiple Equal-Cost Paths Enter the IP Route Table

IP load sharing applies to equal-cost paths in the IP route table. Routes that are eligible for load sharing can enter the table from any of the following sources:

- IP static routes
- Routes learned through RIP

---

[1] IP load sharing is also called "Equal-Cost Multi-Path (ECMP)" load sharing or just "ECMP"

- Routes learned through OSPF
- Routes learned through BGP4

Administrative Distance

The administrative distance is a unique value associated with each type (source) of IP route. Each path has an administrative distance. The administrative distance is not used when performing IP load sharing, but the administrative distance is used when evaluating multiple equal-cost paths to the same destination from different sources, such as RIP, OSPF and so on.

The value of the administrative distance is determined by the source of the route. The Layer 3 Switch is configured with a unique administrative distance value for each IP route source.

When the software receives multiple paths to the same destination and the paths are from different sources, the software compares the administrative distances of the paths and selects the path with the lowest distance. The software then places the path with the lowest administrative distance in the IP route table. For example, if the Layer 3 Switch has a path learned from OSPF and a path learned from RIP for a given destination, only the path with the lower administrative distance enters the IP route table.

Here are the default administrative distances on the Foundry Layer 3 Switch:

- Directly connected – 0 (this value is not configurable)
- Static IP route – 1 (applies to all static routes, including default routes and default network routes)
- Exterior Border Gateway Protocol (EBGP) – 20
- OSPF – 110
- RIP – 120
- Interior Gateway Protocol (IBGP) – 200
- Local BGP – 200
- Unknown – 255 (the router will not use this route)

Lower administrative distances are preferred over higher distances. For example, if the router receives routes for the same network from OSPF and from RIP, the router will prefer the OSPF route by default.

NOTE: You can change the administrative distances individually. See the configuration chapter for the route source for information.

Since the software selects only the path with the lowest administrative distance, and the administrative distance is determined by the path's source, IP load sharing does not apply to paths from different route sources. IP load sharing applies only when the IP route table contains multiple paths to the same destination, from the same IP route source.

IP load sharing does not apply to paths that come from different sources.

Path Cost

The cost parameter provides a common basis of comparison for selecting from among multiple paths to a given destination. Each path in the IP route table has a cost. When the IP route table contains multiple paths to a destination, the Layer 3 Switch chooses the path with the lowest cost. When the IP route table contains more than one path with the lowest cost to a destination, the Layer 3 Switch uses IP load sharing to select one of the lowest-cost paths.

The source of a path's cost value depends on the source of the path.

- IP static route – The value you assign to the metric parameter when you configure the route. The default metric is 1. See “Configuring Load Balancing and Redundancy Using Multiple Static Routes to the Same Destination” on page 29-38.
- RIP – The number of next-hop routers to the destination.
- OSPF – The Path Cost associated with the path. The paths can come from any combination of inter-area, intra-area, and external Link State Advertisements (LSAs).

- BGP4 – The path’s Multi-Exit Discriminator (MED) value.

NOTE: If the path is redistributed between two or more of the above sources before entering the IP route table, the cost can increase during the redistribution due to settings in redistribution filters.

Static Route, OSPF, and BGP4 Load Sharing

IP load sharing and load sharing for static routes, OSPF routes, and BGP4 routes are individually configured. Multiple equal-cost paths for a destination can enter the IP route table only if the source of the paths is configured to support multiple equal-cost paths. For example, if BGP4 allows only one path with a given cost for a given destination, the BGP4 route table cannot contain equal-cost paths to the destination. Consequently, the IP route table will not receive multiple equal-cost paths from BGP4.

Table 29.6 lists the default and configurable maximum numbers of paths for each IP route source that can provide equal-cost paths to the IP route table. The table also lists where to find configuration information for the route source’s load sharing parameters.

The load sharing state for all the route sources is based on the state of IP load sharing. Since IP load sharing is enabled by default on all Foundry Layer 3 Switches, load sharing for static IP routes, RIP routes, OSPF routes, and BGP4 routes also is enabled by default.

Table 29.6: Default Load Sharing Parameters for Route Sources

Route Source	Default Maximum Number of Paths	Maximum Number of Paths	See...
Static IP route	4[1]	6[a]	29-45
RIP	4[a]	6[a]	29-45
OSPF	4	6	29-45
BGP4	1	4	38-21

[1] This value depends on the value for IP load sharing, and is not separately configurable.

How IP Load Sharing Works

When the Layer 3 Switch receives traffic for a destination and the IP route table contains multiple, equal-cost paths to that destination, the device checks the IP forwarding cache for a forwarding entry for the destination. The IP forwarding cache provides a fast path for forwarding IP traffic, including load-balanced traffic. The cache contains entries that associate a destination host or network with a path (next-hop router).

- If the IP forwarding sharing cache contains a forwarding entry for the destination, the device uses the entry to forward the traffic.
- If the IP load forwarding cache does not contain a forwarding entry for the destination, the software selects a path from among the available equal-cost paths to the destination, then creates a forwarding entry in the cache based on the calculation. Subsequent traffic for the same destination uses the forwarding entry.

Response to Path State Changes

If one of the load-balanced paths to a cached destination becomes unavailable, or the IP route table receives a new equal-cost path to a cached destination, the software removes the unavailable path from the IP route table. Then the software selects a new path.

Disabling or Re-Enabling Load Sharing

To disable IP load sharing, enter the following commands:

```
FastIron(config)#no ip load-sharing
```

Syntax: [no] ip load-sharing

Changing the Maximum Number of Load Sharing Paths

By default, IP load sharing allows IP traffic to be balanced across up to four equal paths. You can change the maximum number of paths the Layer 3 Switch supports to a value from 2 – 6.

For optimal results, set the maximum number of paths to a value at least as high as the maximum number of equal-cost paths your network typically contains. For example, if the Layer 3 Switch you are configuring for IP load sharing has six next-hop routers, set the maximum paths value to six.

NOTE: If the setting for the maximum number of paths is lower than the actual number of equal-cost paths, the software does not use all the paths for load sharing.

To change the number of IP load sharing paths, enter a command such as the following:

```
FastIron(config)#ip load-sharing 6
```

Syntax: [no] ip load-sharing [<num>]

The <num> parameter specifies the number of paths and can be from 2 – 6.

Configuring IRDP

The ICMP Router Discovery Protocol (IRDP) is used by Foundry Layer 3 Switches to advertise the IP addresses of its router interfaces to directly attached hosts. IRDP is disabled by default. You can enable the feature on a global basis or on an individual port basis.

- If you enable the feature globally, all ports use the default values for the IRDP parameters.
- If you leave the feature disabled globally but enable it on individual ports, you also can configure the IRDP parameters on an individual port basis.

NOTE: You can configure IRDP parameters only an individual port basis. To do so, IRDP must be disabled globally and enabled only on individual ports. You cannot configure IRDP parameters if the feature is globally enabled.

When IRDP is enabled, the Layer 3 Switch periodically sends Router Advertisement messages out the IP interfaces on which the feature is enabled. The messages advertise the Layer 3 Switch's IP addresses to directly attached hosts who listen for the messages. In addition, hosts can be configured to query the Layer 3 Switch for the information by sending Router Solicitation messages.

Some types of hosts use the Router Solicitation messages to discover their default gateway. When IRDP is enabled on the Foundry Layer 3 Switch, the Layer 3 Switch responds to the Router Solicitation messages. Some clients interpret this response to mean that the Layer 3 Switch is the default gateway. If another router is actually the default gateway for these clients, leave IRDP disabled on the Foundry Layer 3 Switch.

IRDP uses the following parameters. If you enable IRDP on individual ports instead of enabling the feature globally, you can configure these parameters on an individual port basis.

- **Packet type** – The Layer 3 Switch can send Router Advertisement messages as IP broadcasts or as IP multicasts addressed to IP multicast group 224.0.0.1. The packet type is IP broadcast.
- **Maximum message interval and minimum message interval** – When IRDP is enabled, the Layer 3 Switch sends the Router Advertisement messages every 450 – 600 seconds by default. The time within this interval that the Layer 3 Switch selects is random for each message and is not affected by traffic loads or other network factors. The random interval minimizes the probability that a host will receive Router Advertisement messages from other routers at the same time. The interval on each IRDP-enabled Layer 3 Switch interface is independent of the interval on other IRDP-enabled interfaces. The default maximum message interval is 600 seconds. The default minimum message interval is 450 seconds.
- **Hold time** – Each Router Advertisement message contains a hold time value. This value specifies the maximum amount of time the host should consider an advertisement to be valid until a newer advertisement arrives. When a new advertisement arrives, the hold time is reset. The hold time is always longer than the maximum advertisement interval. Therefore, if the hold time for an advertisement expires, the host can reasonably conclude that the router interface that sent the advertisement is no longer available. The default

hold time is three times the maximum message interval.

- **Preference** – If a host receives multiple Router Advertisement messages from different routers, the host selects the router that sent the message with the highest preference as the default gateway. The preference can be a number from -4294967296 to 4294967295. The default is 0.

Enabling IRDP Globally

To globally enable IRDP, enter the following command:

```
FastIron(config)#ip irdp
```

This command enables IRDP on the IP interfaces on all ports. Each port uses the default values for the IRDP parameters. The parameters are not configurable when IRDP is globally enabled.

Enabling IRDP on an Individual Port

To enable IRDP on an individual interface and change IRDP parameters, enter commands such as the following:

```
FastIron(config)#interface ethernet 1/3
FastIron(config-if-1/3)#ip irdp maxadvertinterval 400
```

This example shows how to enable IRDP on a specific port and change the maximum advertisement interval for Router Advertisement messages to 400 seconds.

NOTE: To enable IRDP on individual ports, you must leave the feature globally disabled.

Syntax: [no] ip irdp [broadcast | multicast] [holdtime <seconds>] [maxadvertinterval <seconds>] [minadvertinterval <seconds>] [preference <number>]

The **broadcast | multicast** parameter specifies the packet type the Layer 3 Switch uses to send Router Advertisement.

- **broadcast** – The Layer 3 Switch sends Router Advertisement as IP broadcasts. This is the default.
- **multicast** – The Layer 3 Switch sends Router Advertisement as multicast packets addressed to IP multicast group 224.0.0.1.

The **holdtime** <seconds> parameter specifies how long a host that receives a Router Advertisement from the Layer 3 Switch should consider the advertisement to be valid. When a host receives a new Router Advertisement message from the Layer 3 Switch, the host resets the hold time for the Layer 3 Switch to the hold time specified in the new advertisement. If the hold time of an advertisement expires, the host discards the advertisement, concluding that the router interface that sent the advertisement is no longer available. The value must be greater than the value of the **maxadvertinterval** parameter and cannot be greater than 9000. The default is three times the value of the **maxadvertinterval** parameter.

The **maxadvertinterval** parameter specifies the maximum amount of time the Layer 3 Switch waits between sending Router Advertisements. You can specify a value from 1 to the current value of the **holdtime** parameter. The default is 600 seconds.

The **minadvertinterval** parameter specifies the minimum amount of time the Layer 3 Switch can wait between sending Router Advertisements. The default is three-fourths (0.75) the value of the **maxadvertinterval** parameter. If you change the **maxadvertinterval** parameter, the software automatically adjusts the **minadvertinterval** parameter to be three-fourths the new value of the **maxadvertinterval** parameter. If you want to override the automatically configured value, you can specify an interval from 1 to the current value of the **maxadvertinterval** parameter.

The **preference** <number> parameter specifies the IRDP preference level of this Layer 3 Switch. If a host receives Router Advertisements from multiple routers, the host selects the router interface that sent the message with the highest interval as the host's default gateway. The valid range is -4294967296 to 4294967295. The default is 0.

Configuring RARP

The Reverse Address Resolution Protocol (RARP) provides a simple mechanism for directly-attached IP hosts to boot over the network. RARP allows an IP host that does not have a means of storing its IP address across power cycles or software reloads to query a directly-attached router for an IP address.

RARP is enabled by default. However, you must create a RARP entry for each host that will use the Layer 3 Switch for booting. A RARP entry consists of the following information:

- The entry number – the entry's sequence number in the RARP table.
- The MAC address of the boot client.
- The IP address you want the Layer 3 Switch to give to the client.

When a client sends a RARP broadcast requesting an IP address, the Layer 3 Switch responds to the request by looking in the RARP table for an entry that contains the client's MAC address:

- If the RARP table contains an entry for the client, the Layer 3 Switch sends a unicast response to the client that contains the IP address associated with the client's MAC address in the RARP table.
- If the RARP table does not contain an entry for the client, the Layer 3 Switch silently discards the RARP request and does not reply to the client.

How RARP Differs from BootP/DHCP

RARP and BootP/DHCP are different methods for providing IP addresses to IP hosts when they boot. These methods differ in the following ways:

- Location of configured host addresses
  - RARP requires static configuration of the host IP addresses on the Layer 3 Switch. The Layer 3 Switch replies directly to a host's request by sending an IP address you have configured in the RARP table.
  - The Layer 3 Switch forwards BootP and DHCP requests to a third-party BootP/DHCP server that contains the IP addresses and other host configuration information.
- Connection of host to boot source (Layer 3 Switch or BootP/DHCP server):
  - RARP requires the IP host to be directly attached to the Layer 3 Switch.
  - An IP host and the BootP/DHCP server can be on different networks and on different routers, so long as the routers are configured to forward ("help") the host's boot request to the boot server.
  - You can centrally configure other host parameters on the BootP/DHCP server, in addition to the IP address, and supply those parameters to the host along with its IP address.

To configure the Layer 3 Switch to forward BootP/DHCP requests when boot clients and the boot servers are on different subnets on different Layer 3 Switch interfaces, see "Configuring BootP/DHCP Relay Parameters" on page 29-50.

Disabling RARP

RARP is enabled by default. To disable RARP, enter the following command at the global CONFIG level:

```
FastIron(config)#no ip rarp
```

Syntax: [no] ip rarp

To re-enable RARP, enter the following command:

```
FastIron(config)#ip rarp
```

Creating Static RARP Entries

You must configure the RARP entries for the RARP table. The Layer 3 Switch can send an IP address in reply to a client's RARP request only if create a RARP entry for that client.

To assign a static IP RARP entry for static routes on a Foundry router, enter a command such as the following:

```
FastIron(config)#rarp 1 1245.7654.2348 192.53.4.2
```

This command creates a RARP entry for a client with MAC address 1245.7654.2348. When the Layer 3 Switch receives a RARP request from this client, the Layer 3 Switch replies to the request by sending IP address 192.53.4.2 to the client.

Syntax: rarp <number> <mac-addr>.<ip-addr>

The <number> parameter identifies the RARP entry number. You can specify an unused number from 1 to the maximum number of RARP entries supported on the device. To determine the maximum number of entries supported on the device, see the section "Displaying and Modifying System Parameter Default Settings" on page 8-11.

The <mac-addr> parameter specifies the MAC address of the RARP client.

The <ip-addr> parameter specifies the IP address the Layer 3 Switch will give the client in response to the client's RARP request.

Changing the Maximum Number of Static RARP Entries Supported

The number of RARP entries the Layer 3 Switch supports depends on how much memory the Layer 3 Switch has. To determine how many RARP entries your Layer 3 Switch can have, display the system default information using the procedure in the section "Displaying and Modifying System Parameter Default Settings" on page 8-11.

If your Layer 3 Switch allows you to increase the maximum number of RARP entries, you can use a procedure in the same section to do so.

NOTE: You must save the configuration to the startup-config file and reload the software after changing the RARP cache size to place the change into effect.

Configuring UDP Broadcast and IP Helper Parameters

Some applications rely on client requests sent as limited IP broadcasts addressed to the UDP's application port. If a server for the application receives such a broadcast, the server can reply to the client. Routers do not forward subnet directed broadcasts, so the client and server must be on the same network for the broadcast to reach the server. If the client and server are on different networks (on opposite sides of a router), the client's request cannot reach the server.

You can configure the Layer 3 Switch to forward clients‘ requests to UDP application servers. To do so:

- Enable forwarding support for the UDP application port, if forwarding support is not already enabled.
- Configure a helper adders on the interface connected to the clients. Specify the helper address to be the IP address of the application server or the subnet directed broadcast address for the IP subnet the server is in. A helper address is associated with a specific interface and applies only to client requests received on that interface. The Layer 3 Switch forwards client requests for any of the application ports the Layer 3 Switch is enabled to forward to the helper address.

Forwarding support for the following application ports is enabled by default.

- bootps (port 67)
- dns (port 53)
- tftp (port 69)
- time (port 37)
- netbios-ns (port 137)
- netbios-dgm (port 138)
- tacacs (port 65)

NOTE: The application names are the names for these applications that the Layer 3 Switch software recognizes, and might not match the names for these applications on some third-party devices. The numbers listed in parentheses are the UDP port numbers for the applications. The numbers come from RFC 1340.

NOTE: Forwarding support for BootP/DHCP is enabled by default. If you are configuring the Layer 3 Switch to forward BootP/DHCP requests, see “Configuring BootP/DHCP Relay Parameters” on page 29-50.

You can enable forwarding for other applications by specifying the application port number.

You also can disable forwarding for an application.

NOTE: If you disable forwarding for a UDP application, forwarding of client requests received as broadcasts to helper addresses is disabled. Disabling forwarding of an application does not disable other support for the application. For example, if you disable forwarding of Telnet requests to helper addresses, other Telnet support on the Layer 3 Switch is not also disabled.

Enabling Forwarding for a UDP Application

If you want the Layer 3 Switch to forward client requests for UDP applications that the Layer 3 Switch does not forward by default, you can enable forwarding support for the port. To enable forwarding support for a UDP application, use the following method. You also can disable forwarding for an application using this method.

NOTE: You also must configure a helper address on the interface that is connected to the clients for the application. The Layer 3 Switch cannot forward the requests unless you configure the helper address. See “Configuring an IP Helper Address” on page 29-51.

To enable the forwarding of SNMP trap broadcasts, enter the following command:

```
FastIron(config)#ip forward-protocol udp snmp-trap
```

Syntax: [no] ip forward-protocol udp <udp-port-name> | <udp-port-num>

The <udp-port-name> parameter can have one of the following values. For reference, the corresponding port numbers from RFC 1340 are shown in parentheses. If you specify an application name, enter the name only, not the parentheses or the port number shown here.

- bootpc (port 68)
- bootps (port 67)
- discard (port 9)
- dns (port 53)
- dnsix (port 90)
- echo (port 7)
- mobile-ip (port 434)
- netbios-dgm (port 138)
- netbios-ns (port 137)
- ntp (port 123)
- tacacs (port 65)
- talk (port 517)
- time (port 37)
- tftp (port 69)

In addition, you can specify any UDP application by using the application’s UDP port number.

The <udp-port-num> parameter specifies the UDP application port number. If the application you want to enable is not listed above, enter the application port number. You also can list the port number for any of the applications listed above.

To disable forwarding for an application, enter a command such as the following:

```
FastIron(config)#no ip forward-protocol udp snmp
```

This command disables forwarding of SNMP requests to the helper addresses configured on Layer 3 Switch interfaces.

Configuring an IP Helper Address

To forward a client's broadcast request for a UDP application when the client and server are on different networks, you must configure a helper address on the interface connected to the client. Specify the server's IP address or the subnet directed broadcast address of the IP subnet the server is in as the helper address.

You can configure up to 16 helper addresses on each interface. You can configure a helper address on an Ethernet port or a virtual interface.

To configure a helper address on interface 2 on chassis module 1, enter the following commands:

```
FastIron(config)#interface e 1/2
FastIron(config-if-1/2)#ip helper-address 1 207.95.7.6
```

The commands in this example change the CLI to the configuration level for port 1/2, then add a helper address for server 207.95.7.6 to the port. If the port receives a client request for any of the applications that the Layer 3 Switch is enabled to forward, the Layer 3 Switch forwards the client's request to the server.

Syntax: ip helper-address <num> <ip-addr>

The <num> parameter specifies the helper address number and can be from 1 – 16.

The <ip-addr> command specifies the server's IP address or the subnet directed broadcast address of the IP subnet the server is in.

Configuring BootP/DHCP Relay Parameters

A host on an IP network can use BootP/DHCP to obtain its IP address from a BootP/DHCP server. To obtain the address, the client sends a BootP/DHCP request. The request is a subnet directed broadcast and is addressed to UDP port 67. A limited IP broadcast is addressed to IP address 255.255.255.255 and is not forwarded by the Foundry Layer 3 Switch or other IP routers.

When the BootP/DHCP client and server are on the same network, the server receives the broadcast request and replies to the client. However, when the client and server are on different networks, the server does not receive the client's request, because the Layer 3 Switch does not forward the request.

You can configure the Layer 3 Switch to forward BootP/DHCP requests. To do so, configure a helper address on the interface that receives the client requests, and specify the BootP/DHCP server's IP address as the address you are helping the BootP/DHCP requests to reach. Instead of the server's IP address, you can specify the subnet directed broadcast address of the IP subnet the server is in.

BootP/DHCP Relay Parameters

The following parameters control the Layer 3 Switch's forwarding of BootP/DHCP requests:

- **Helper address** – The BootP/DHCP server's IP address. You must configure the helper address on the interface that receives the BootP/DHCP requests from the client. The Layer 3 Switch cannot forward a request to the server unless you configure a helper address for the server.
- **Gateway address** – The Layer 3 Switch places the IP address of the interface that received the BootP/DHCP request in the request packet's Gateway Address field (sometimes called the Router ID field). When the server responds to the request, the server sends the response as a unicast packet to the IP address in the Gateway Address field. (If the client and server are directly attached, the Gateway ID field is empty and the server replies to the client using a unicast or broadcast packet, depending on the server.)

  By default, the Layer 3 Switch uses the lowest-numbered IP address on the interface that receives the request as the Gateway address. You can override the default by specifying the IP address you want the Layer 3 Switch to use.
- **Hop Count** – Each router that forwards a BootP/DHCP packet increments the hop count by 1. Routers also discard a forwarded BootP/DHCP request instead of forwarding the request if the hop count is greater than the maximum number of BootP/DHCP hops allows by the router. By default, a Foundry Layer 3 Switch

forwards a BootP/DHCP request if its hop count is four or less, but discards the request if the hop count is greater than four. You can change the maximum number of hops the Layer 3 Switch will allow to a value from 1 – 15.

NOTE: The BootP/DHCP hop count is not the TTL parameter.

Configuring an IP Helper Address

The procedure for configuring a helper address for BootP/DHCP requests is the same as the procedure for configuring a helper address for other types of UDP broadcasts. See “Configuring an IP Helper Address” on page 29-50.

Configuring the BOOTP/DHCP Reply Source Address

Platform Support:

- FESX/FSX/FWSX devices running software release 03.0.00 and later

You can configure the Foundry device so that a BOOTP/DHCP reply to a client contains the server’s IP address as the source address instead of the router’s IP address. To do so, enter the following command at the Global CONFIG level of the CLI:

```
FastIron(config)#ip helper-use-responder-ip
```

Syntax: [no] ip helper-use-responder-ip

Changing the IP Address Used for Stamping BootP/DHCP Requests

When the Layer 3 Switch forwards a BootP/DHCP request, the Layer 3 Switch “stamps” the Gateway Address field. The default value the Layer 3 Switch uses to stamp the packet is the lowest-numbered IP address configured on the interface that received the request. If you want the Layer 3 Switch to use a different IP address to stamp requests received on the interface, use either of the following methods to specify the address.

The BootP/DHCP stamp address is an interface parameter. Change the parameter on the interface that is connected to the BootP/DHCP client.

To change the IP address used for stamping BootP/DHCP requests received on interface 1/1, enter commands such as the following:

```
FastIron(config)#int e 1/1
FastIron(config-if-1/1)#ip bootp-gateway 109.157.22.26
```

These commands change the CLI to the configuration level for port 1/1, then change the BootP/DHCP stamp address for requests received on port 1/1 to 192.157.22.26. The Layer 3 Switch will place this IP address in the Gateway Address field of BootP/DHCP requests that the Layer 3 Switch receives on port 1/1 and forwards to the BootP/DHCP server.

Syntax: ip bootp-gateway <ip-addr>

Changing the Maximum Number of Hops to a BootP Relay Server

Each BootP/DHCP request includes a field Hop Count field. The Hop Count field indicates how many routers the request has passed through. When the Layer 3 Switch receives a BootP/DHCP request, the Layer 3 Switch looks at the value in the Hop Count field.

- If the hop count value is equal to or less than the maximum hop count the Layer 3 Switch allows, the Layer 3 Switch increments the hop count by one and forwards the request.
- If the hop count is greater than the maximum hop count the Layer 3 Switch allows, the Layer 3 Switch discards the request.

To change the maximum number of hops the Layer 3 Switch allows for forwarded BootP/DHCP requests, use either of the following methods.

NOTE: The BootP/DHCP hop count is not the TTL parameter.

To modify the maximum number of BootP/DHCP hops, enter the following command:

```
FastIron(config)#bootp-relay-max-hops 10
```

This command allows the Layer 3 Switch to forward BootP/DHCP requests that have passed through ten previous hops before reaching the Layer 3 Switch. Requests that have traversed 11 hops before reaching the switch are dropped. Since the hop count value initializes at zero, the hop count value of an ingressing DHCP Request packet is the number of Layer 3 routers that the packet has already traversed.

Syntax: bootp-relay-max-hops <1 through 15>

DHCP Client-Based Auto-Configuration

Platform Support: FGS and FLS devices running software release 4.2.00 or later

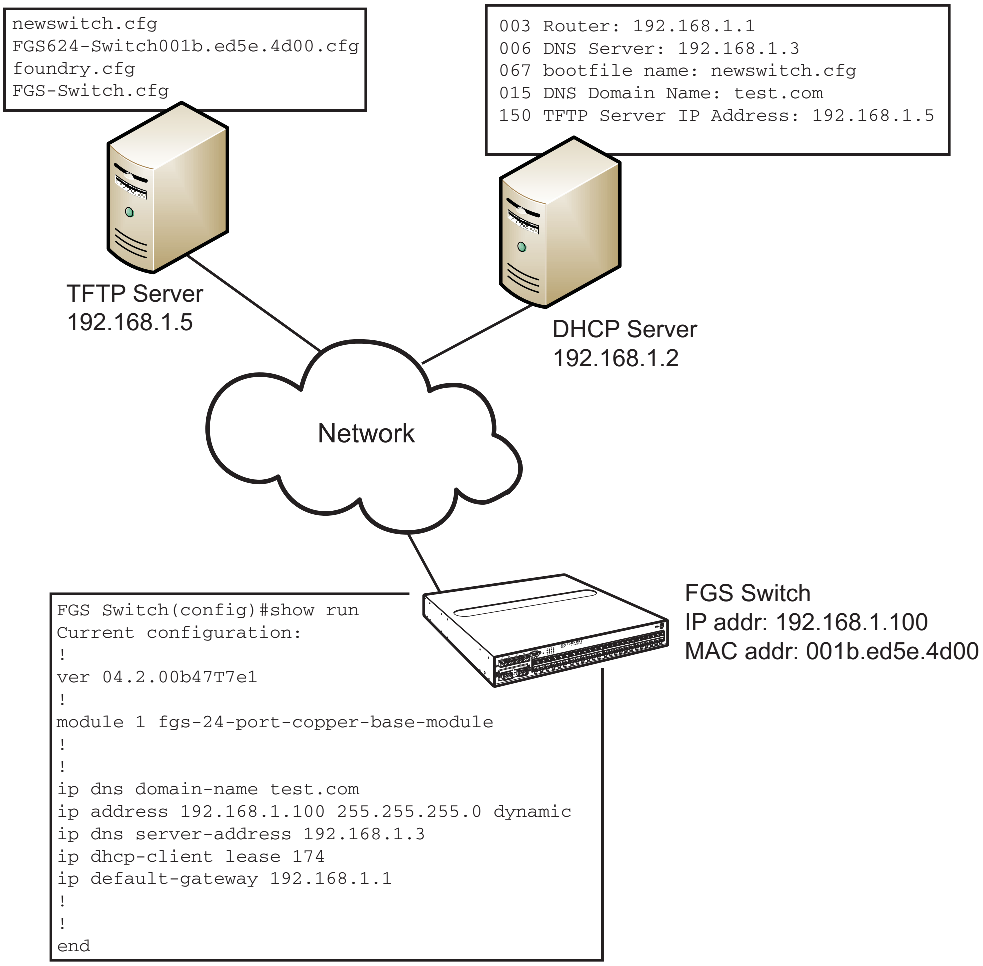
DHCP Client-Based Auto-Configuration allows FGS or FLS Layer 2 and Base Layer 3 devices to automatically obtain leased IP addresses through a DHCP server, negotiate address lease renewal, and obtain a configuration file.

DHCP Client-Based Auto-Configuration occurs in two steps:

- In the first step, the IP address validation and lease negotiation enables the DHCP client (an FGS or FLS Layer 2 or Base-Layer 3 device) to automatically obtain and configure an IP address, as follows:
  - One lease is granted for each Layer 2 device. if the device is configured with a static IP address, the DHCP Auto-Configuration feature is automatically disabled.
  - For a Base Layer 3 devices, one leased address is granted (per device) to the interface that first receives a response from the DHCP server.
- In the second step, TFTP configuration download and update, the device downloads a configuration file from a TFTP server and saves it as the running configuration.

Figure 29.6 shows how DHCP Client-Based Auto Configuration works.

Figure 29.6 DHCP Client-Based Auto Configuration



```
newswitch.cfg
FGS624-Switch001b.ed5e.4d00.cfg
foundry.cfg
FGS-Switch.cfg
```

```
003 Router: 192.168.1.1
006 DNS Server: 192.168.1.3
067 bootfile name: newswitch.cfg
015 DNS Domain Name: test.com
150 TFTP Server IP Address: 192.168.1.5
```

```
FGS Switch(config)#show run
Current configuration:
!
ver 04.2.00b47T7e1
!
module 1 fgs-24-port-copper-base-module
!
!
ip dns domain-name test.com
ip address 192.168.1.100 255.255.255.0 dynamic
ip dns server-address 192.168.1.3
ip dhcp-client lease 174
ip default-gateway 192.168.1.1
!
!
end
```

How DHCP Client-Based Auto-Configuration Works

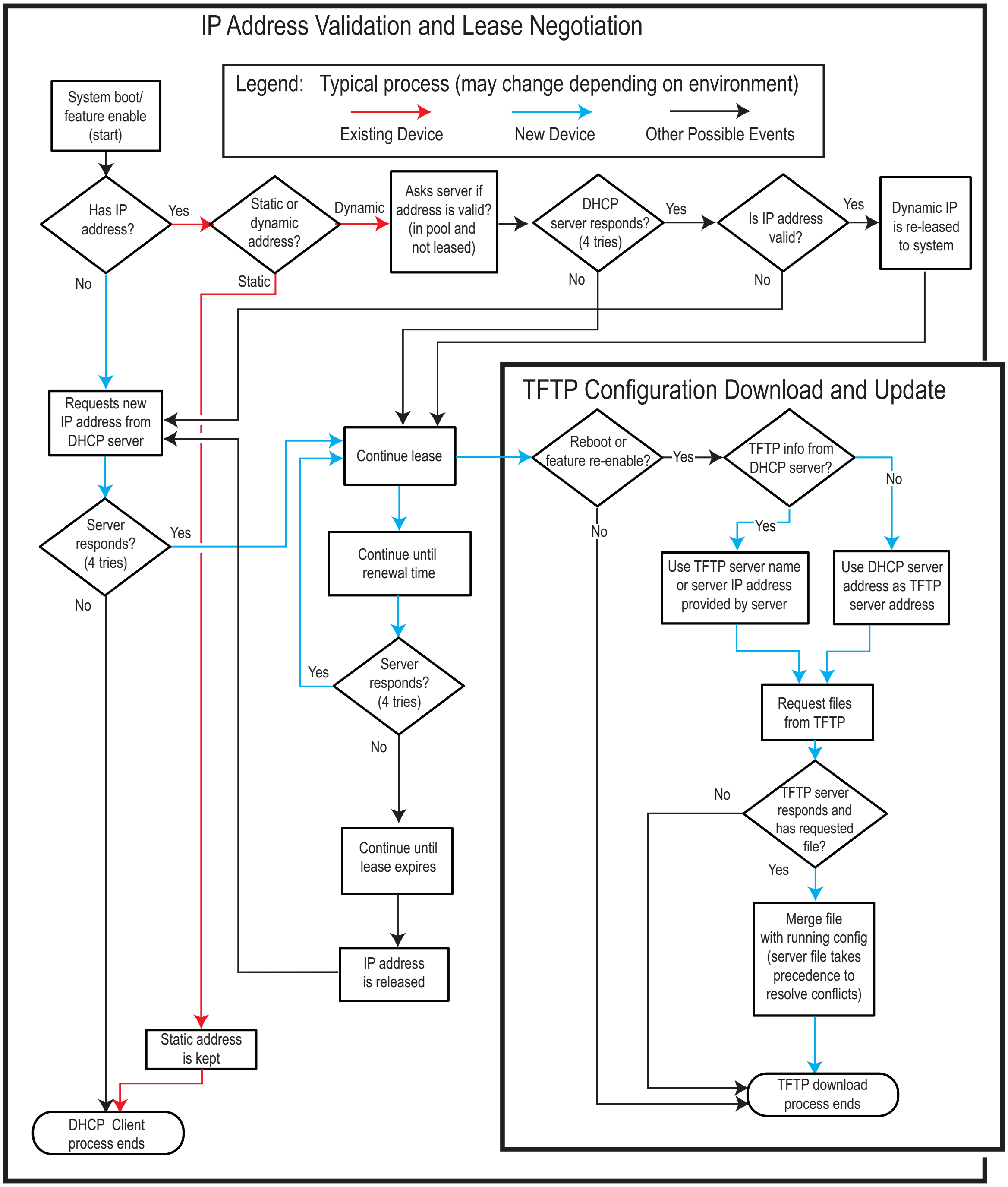
Auto-Configuration is enabled by default. To disable this feature, see “Disabling or Re-Enabling Auto-Configuration” on page 29-56.

The steps of the Auto-Configuration process are described in Figure 29.7, and in the description that follows the flowchart.

NOTE: For Base Layer 3 devices, this feature is available for the default VLAN only. For Layer 2 devices, This feature is available for default VLANs and management VLANs. This feature is not supported on virtual interfaces (VEs), trunked ports, or LACP ports.

NOTE: Although the DHCP server may provide multiple addresses, only one IP address is installed at a time.

Figure 29.7 The DHCP Client-Based Auto-Configuration Steps


IP Address Validation and Lease Negotiation
Legend: Typical process (may change depending on environment)
Existing Device
New Device
Other Possible Events
System boot/ feature enable (start)
Has IP address?
Yes
No
Static or dynamic address?
Dynamic
Static
Asks server if address is valid? (in pool and not leased)
DHCP server responds? (4 tries)
Yes
No
Is IP address valid?
Yes
No
Dynamic IP is re-leased to system
Requests new IP address from DHCP server
Server responds? (4 tries)
Yes
No
Continue lease
Continue until renewal time
Server responds? (4 tries)
Yes
No
Continue until lease expires
IP address is released
Static address is kept
DHCP Client process ends
TFTP Configuration Download and Update
Reboot or feature re-enable?
Yes
No
TFTP info from DHCP server?
Yes
No
Use TFTP server name or server IP address provided by server
Use DHCP server address as TFTP server address
Request files from TFTP
TFTP server responds and has requested file?
No
Yes
Merge file with running config (server file takes precedence to resolve conflicts)
TFTP download process ends

The IP Address Validation and Lease Negotiation Step

1. At boot-up, the device automatically checks its configuration for an IP address.
2. If the device ***does not have*** a static IP address, it requests the lease of an address from the DHCP server.
   - If the server responds, it leases an IP address to the device for the specified lease period.
   - If the server does not respond (after four tries) the DHCP Client process is ended.

3. If the device ***has*** a ***dynamic*** address, the device asks the DHCP server to validate that address. If the server does not respond, the device will continue to use the existing address until the lease expires. If the server responds, and the IP address is outside of the DHCP address pool or has been leased to another device, it is automatically rejected, and the device receives a new IP address from the server. If the existing address is valid, the lease continues.

NOTE: The lease time interval is configured on the DHCP server, not on the client device. The **ip dhcp-client lease** command is set by the system, and is non-operational to a user.

4. If the existing address is **static**, the device keeps it and the DHCP Client process is ended.
5. For a leased IP address, when the lease interval reaches the renewal point, the device requests a renewal from the DHCP server.
   - If the device is ***able*** to contact the DHCP server at the renewal point in the lease, the DHCP server extends the lease. This process can continue indefinitely.
   - If the device is ***unable*** to reach the DHCP server after four attempts, it continues to use the existing IP address until the lease expires. When the lease expires, the dynamic IP address is removed and the device contacts the DHCP server for a new address. If the device is still unable to contact the DHCP server after four attempts, the process is ended.

The TFTP Configuration Download and Update Step

NOTE: This process only occurs when the client device reboots, or when Auto-Configuration has been disabled and then re-enabled.

1. When the device reboots, or the Auto-Configuration feature has been disabled and then re-enabled, the device uses information from the DHCP server to contact the TFTP server to update the running-configuration file.
   - If the DHCP server provides a TFTP server name or IP address, the device uses this information to request files from the TFTP server.
   - If the DHCP server does not provide a TFTP server name or IP address, the device requests the configuration files from the DHCP server.
2. The device requests the configuration files from the TFTP server by asking for filenames in the following order:
   - **bootfile name provided by the DHCP server** (if configured)
   - **hostnameMAC-config.cfg,** for example:

   `FGS624p-Switch001b.ed5e.4d00-config.cfg`

   - **hostnameMAC.cfg,** for example:

   `FGS624p-Switch001b.ed5e.4d00.cfg`

   - **foundry.cfg** (applies to both FGS and FLS devices), for example:

   `foundry.cfg`

   - **<fgs | fls>-<switch | router>.cfg** (applies to Layer 2 or Base Layer 3 devices), for example:

`fgs-switch.cfg`	(FGS Layer 2)
`fls-switch.cfg`	(FLS Layer 2)
`fgs-router.cfg`	(FGS Base-Layer 3)
`fls-router.cfg`	(FLS Base Layer 3)

   If the device is successful in contacting the TFTP server and the server has the configuration file, the files are merged. If there is a conflict, the server file takes precedence.

   If the device is ***unable*** to contact the TFTP server or if the files are not found on the server, the TFTP part of the configuration download process ends.

Supported Options for DHCP Servers

DHCP Client-Based Auto-Configuration supports the following options:

- 001 - subnetmask
- 003 - router ip
- 015 - domain name
- 006 - domain name server
- 012 - hostname (optional)
- 066 - TFTP server name
- 067 - bootfile name
- 150 - TFTP server IP address (private option, datatype = IP Address)

Disabling or Re-Enabling Auto-Configuration

You can disable or enable this feature using the following commands:

For a switch:

```
FGS Switch(config)#ip dhcp-client enable
FGS Switch(config)#no ip dhcp-client enable
```

For a router:

```
FGS Router(config-if-e1000-0/1/1)#ip dhcp-client enable
FGS Router(config-if-e1000-0/1/1)#no ip dhcp-client enable
```

Syntax: [no] ip dhcp-client enable

Configuration Notes

- When using DHCP on a router, if you have a DHCP address for one interface, and you want to connect to the DHCP server from another interface, you must disable DHCP on the first interface, then enable DHCP on the second interface.
- When DHCP is disabled, and then re-enabled, or if the system is rebooted, the TFTP process requires approximately three minutes to run in the background before file images can be downloaded manually.
- Once a port is assigned a leased IP address, it is bound by the terms of the lease regardless of the link state of the port.

Displaying DHCP Configuration Information

The following example shows output from the **show ip** command for Layer 2 devices):

```
FGS Switch(config)#show ip

     Switch IP address: 10.44.16.116

           Subnet mask: 255.255.255.0

Default router address: 10.44.16.1
   TFTP server address: 10.44.16.41
Configuration filename: foundry.cfg
        Image filename: None
```

The following example shows output from the show ip address command for a Layer 2 device:

```
FGS Switch(config)#show ip address
  IP Address       Type      Lease Time       Interface
10.44.16.116       Dynamic    174             0/1/1
```

The following example shows output from the **show ip address** command for a Base Layer 3 device:

```
FGS Router(config)#show ip address
  IP Address       Type      Lease Time       Interface
 10.44.3.233       Dynamic   672651           0/1/2
     1.0.0.1         Static    N/A                0/1/15
```

The following example shows a Layer 2 device configuration as a result of the **show run** command:

```
FGS Switch(config)#show run
Current configuration:
!
ver 04.2.00b47T7e1
!
module 1 fls-24-port-copper-base-module
!
!
ip address 10.44.16.116 255.255.255.0 dynamic
ip dns server-address 10.44.16.41
ip dhcp-client lease 174
ip default-gateway 10.44.16.1
!
!
end
```

The following example shows a Base Layer 3 device configuration as a result of the **show run** command:

```
FGS Router(config)#show run
Current configuration:
!
ver 04.2.00b47T7e1
!
module 1 fgs-24-port-management-module
module 2 fgs-cx4-2-port-10g-module
module 3 fgs-xfp-1-port-10g-module
!
vlan 1 name DEFAULT-VLAN by port
!
ip dns domain-name test.com
ip dns server-address 10.44.3.111
interface ethernet 0/1/2
 ip address 10.44.3.233 255.255.255.0 dynamic
 ip dhcp-client lease 691109
!
interface ethernet 0/1/15
 ip address 1.0.0.1 255.0.0.0
 ip helper-address 1 10.44.3.111
!
end
```

DHCP Log Messages

The following DHCP notification messages are sent to the log file:

```
2d01h48m21s:I: DHCPC: existing ip address found, no further action needed by DHCPC
2d01h48m21s:I: DHCPC: Starting DHCP Client service
2d01h48m21s:I: DHCPC: Stopped DHCP Client service
2d01h48m21s:I: DHCPC: FGS624P Switch running-configuration changed
2d01h48m21s:I: DHCPC: sending TFTP request for bootfile name fgs-switch.cfg
2d01h48m21s:I: DHCPC: TFTP unable to download running-configuration
2d01h48m21s:I: DHCPC: Found static IP Address 1.1.1.1 subnet mask 255.255.255.0 on
port 0/1/5
2d01h48m21s:I: DHCPC: Client service found no DHCP server(s) on 3 possible subnet
2d01h48m21s:I: DHCPC: changing 0/1/3 protocol from stopped to running
```

Configuring IP Parameters – Layer 2 Switches

The following sections describe how to configure IP parameters on a Foundry Layer 2 Switch.

NOTE: This section describes how to configure IP parameters for Layer 2 Switches. For IP configuration information for Layer 3 Switches, see “Configuring IP Parameters – Layer 3 Switches” on page 29-16.

Configuring the Management IP Address and Specifying the Default Gateway

To manage a Layer 2 Switch using Telnet or Secure Shell (SSH) CLI connections or the Web management interface, you must configure an IP address for the Layer 2 Switch. Optionally, you also can specify the default gateway.

Foundry devices support both classical IP network masks (Class A, B, and C subnet masks, and so on) and Classless Interdomain Routing (CIDR) network prefix masks.

- To enter a classical network mask, enter the mask in IP address format. For example, enter “209.157.22.99 255.255.255.0” for an IP address with a Class-C subnet mask.
- To enter a prefix network mask, enter a forward slash (/) and the number of bits in the mask immediately after the IP address. For example, enter “209.157.22.99/24” for an IP address that has a network mask with 24 significant bits (ones).

By default, the CLI displays network masks in classical IP address format (example: 255.255.255.0). You can change the display to prefix format. See “Changing the Network Mask Display to Prefix Format” on page 29-64.

To assign an IP address to a Foundry Layer 2 Switch, enter a command such as the following at the global CONFIG level:

```
FastIron(config)#ip address 192.45.6.110 255.255.255.0
```

Syntax: ip address <ip-addr> <ip-mask>

or

Syntax: ip address <ip-addr>/<mask-bits>

NOTE: You also can enter the IP address and mask in CIDR format, as follows:

```
FastIron(config)#ip address 192.45.6.1/24
```

To specify the Layer 2 Switch’s default gateway, enter a command such as the following:

```
FastIron(config)#ip default-gateway 192.45.6.1 255.255.255.0
```

Syntax: ip default-gateway <ip-addr>

or

Syntax: ip default-gateway <ip-addr>/<mask-bits>

Configuring Domain Name Server (DNS) Resolver

The Domain Name Server (DNS) resolver feature lets you use a host name to perform Telnet, ping, and traceroute commands. You can also define a DNS domain on a Foundry Layer 2 Switch or Layer 3 Switch and thereby recognize all hosts within that domain. After you define a domain name, the Foundry Layer 2 Switch or Layer 3 Switch automatically appends the appropriate domain to the host and forwards it to the domain name server.

For example, if the domain “newyork.com” is defined on a Foundry Layer 2 Switch or Layer 3 Switch and you want to initiate a ping to host “NYC01” on that domain, you need to reference only the host name in the command instead of the host name and its domain name. For example, you could enter either of the following commands to initiate the ping:

```
FastIron#ping nyc01
FastIron#ping nyc01.newyork.com
```

Defining a DNS Entry

You can define up to four DNS servers for each DNS entry. The first entry serves as the primary default address. If a query to the primary address fails to be resolved after three attempts, the next gateway address is queried (also up to three times). This process continues for each defined gateway address until the query is resolved. The order in which the default gateway addresses are polled is the same as the order in which you enter them.

Suppose you want to define the domain name of newyork.com on a Layer 2 Switch and then define four possible default DNS gateway addresses. To do so, enter the following commands:

```
FastIron(config)#ip dns domain-name newyork.com
FastIron(config)#ip dns server-address 209.157.22.199 205.96.7.15 208.95.7.25
201.98.7.15
```

Syntax: ip dns server-address <ip-addr> [<ip-addr>] [<ip-addr>] [<ip-addr>]

In this example, the first IP address in the **ip dns server-address...** command becomes the primary gateway address and all others are secondary addresses. Because IP address 201.98.7.15 is the last address listed, it is also the last address consulted to resolve a query.

Using a DNS Name To Initiate a Trace Route

EXAMPLES:

Suppose you want to trace the route from a Foundry Layer 2 Switch to a remote server identified as NYC02 on domain newyork.com. Because the newyork.com domain is already defined on the Layer 2 Switch, you need to enter only the host name, NYC02, as noted below.

```
FastIron#traceroute nyc02
```

Syntax: traceroute <host-ip-addr> [maxttl <value>] [minttl <value>] [numeric] [timeout <value>] [source-ip <ip addr>]

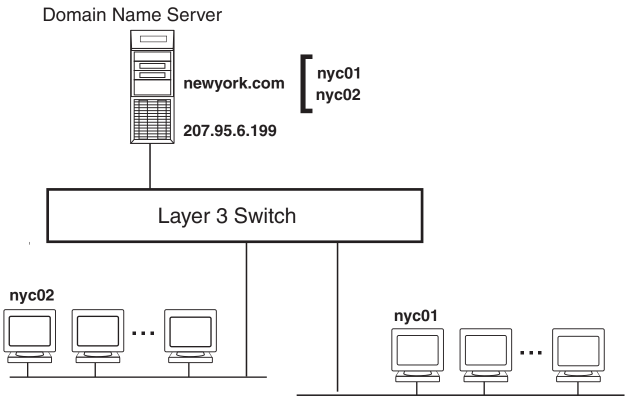
The only required parameter is the IP address of the host at the other end of the route.

After you enter the command, a message indicating that the DNS query is in process and the current gateway address (IP address of the domain name server) being queried appear on the screen:

```
Type Control-c to abort
Sending DNS Query to 209.157.22.199
Tracing Route to IP node 209.157.22.80
To ABORT Trace Route, Please use stop-traceroute command.
 Traced route to target IP node 209.157.22.80:
   IP Address          Round Trip Time1    Round Trip Time2
  207.95.6.30         93 msec             121 msec
```

NOTE: In the above example, 209.157.22.199 is the IP address of the domain name server (default DNS gateway address), and 209.157.22.80 represents the IP address of the NYC02 host.

Figure 29.8 Querying a Host on the newyork.com Domain



Changing the TTL Threshold

The TTL threshold prevents routing loops by specifying the maximum number of router hops an IP packet originated by the Layer 2 Switch can travel through. Each device capable of forwarding IP that receives the packet decrements (decreases) the packet's TTL by one. If a router receives a packet with a TTL of 1 and reduces the TTL to zero, the router drops the packet.

The default TTL is 64. You can change the TTL to a value from 1 – 255.

To modify the TTL threshold to 25, enter the following commands:

```
FastIron(config)#ip ttl 25
FastIron(config)#exit
```

Syntax: ip ttl <1-255>

Configuring DHCP Assist

Platform Support:

- FESX/FSX/FWSX devices – all software releases
- FGS and FLS devices running software release 03.0.00 and later

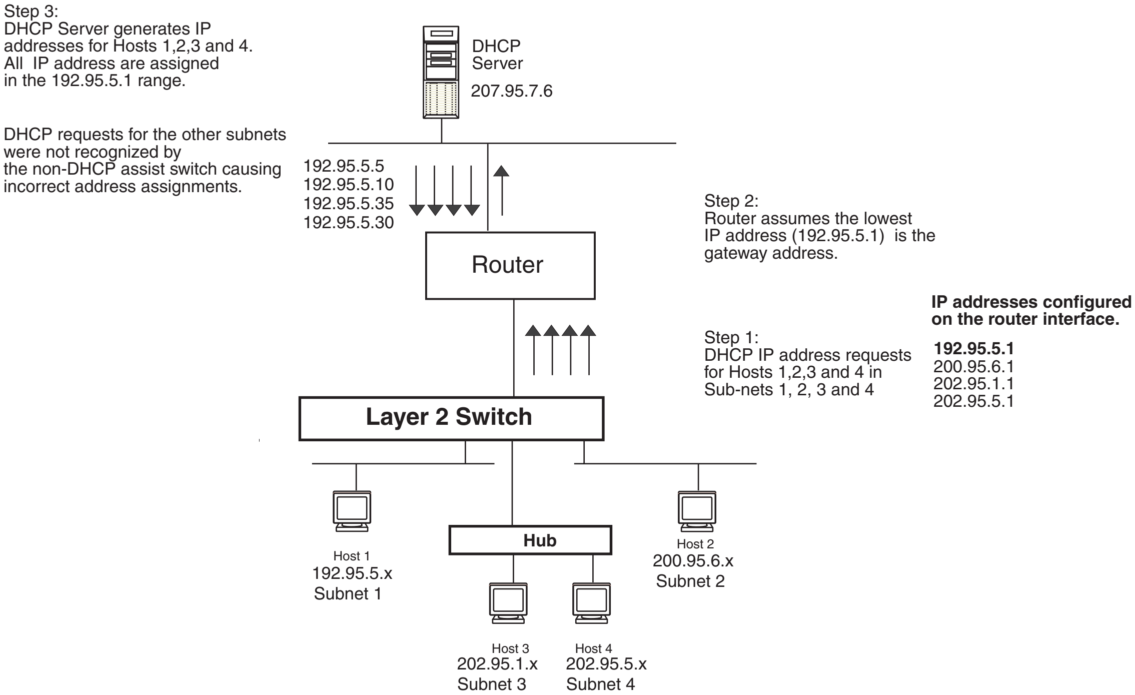
DHCP Assist allows a Foundry Layer 2 Switch to assist a router that is performing multi-netting on its interfaces as part of its DHCP relay function.

DHCP Assist ensures that a DHCP server that manages multiple IP subnets can readily recognize the requester's IP subnet, even when that server is not on the client's local LAN segment. The Foundry Layer 2 Switch does so by stamping each request with its IP gateway address in the DHCP discovery packet.

NOTE: Foundry Layer 3 Switches provide BootP/DHCP assistance by default on an individual port basis. See "Changing the IP Address Used for Stamping BootP/DHCP Requests" on page 29-51.

By allowing multiple subnet DHCP requests to be sent on the same wire, you can reduce the number of router ports required to support secondary addressing as well as reduce the number of DHCP servers required, by allowing a server to manage multiple subnet address assignments.

Figure 29.9 DHCP Requests in a Network without DHCP Assist on the Layer 2 Switch


Step 3:
DHCP Server generates IP addresses for Hosts 1,2,3 and 4. All IP address are assigned in the 192.95.5.1 range.
DHCP requests for the other subnets were not recognized by the non-DHCP assist switch causing incorrect address assignments.
DHCP Server
207.95.7.6
192.95.5.5
192.95.5.10
192.95.5.35
192.95.5.30
Router
Step 2:
Router assumes the lowest IP address (192.95.5.1) is the gateway address.
IP addresses configured on the router interface.
192.95.5.1
200.95.6.1
202.95.1.1
202.95.5.1
Step 1:
DHCP IP address requests for Hosts 1,2,3 and 4 in Sub-nets 1, 2, 3 and 4
Layer 2 Switch
Hub
Host 1
192.95.5.x
Subnet 1
Host 2
200.95.6.x
Subnet 2
Host 3
202.95.1.x
Subnet 3
Host 4
202.95.5.x
Subnet 4

In a network operating without DHCP Assist, hosts can be assigned IP addresses from the wrong subnet range because a router with multiple subnets configured on an interface cannot distinguish among DHCP discovery packets received from different subnets.

For example, in Figure 29.9, a host from each of the four subnets supported on a Layer 2 Switch requests an IP address from the DHCP server. These requests are sent transparently to the router. Because the router is unable to determine the origin of each packet by subnet, it assumes the lowest IP address or the 'primary address' is the gateway for all ports on the Layer 2 Switch and stamps the request with that address.

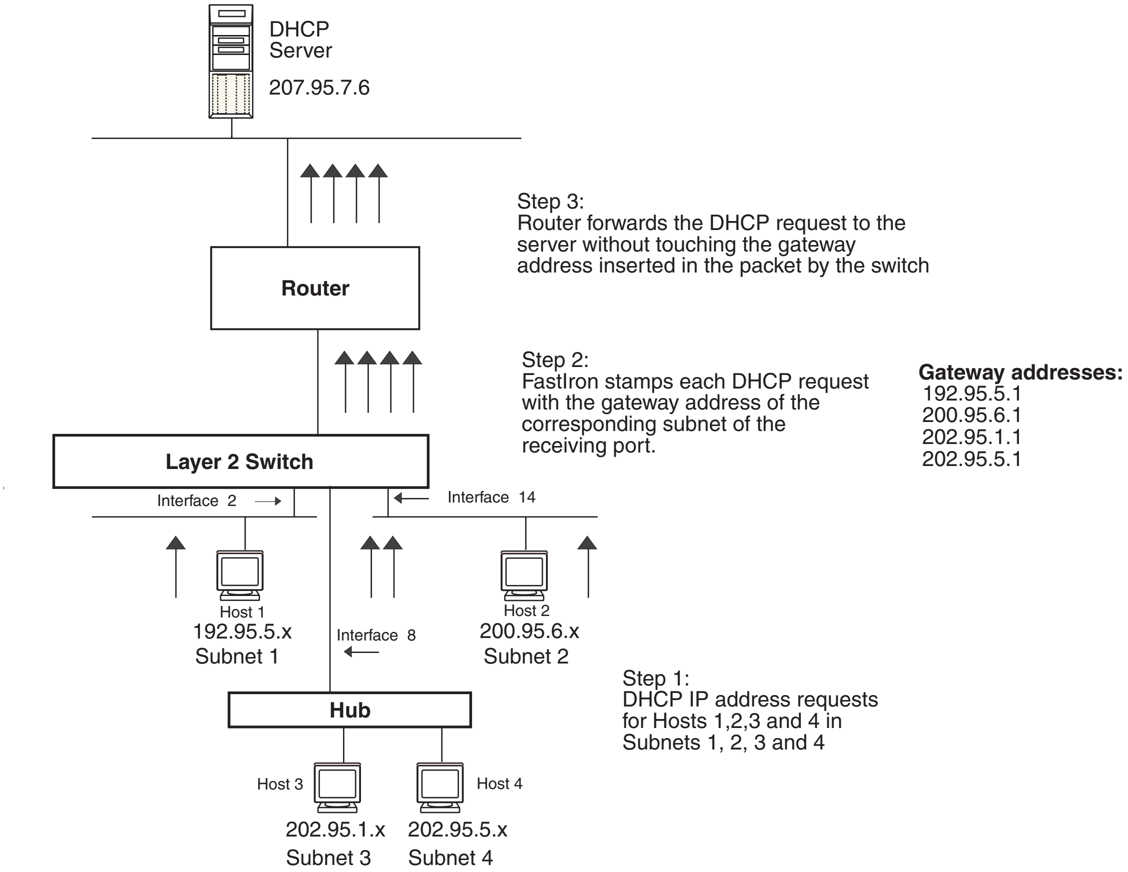
When the DHCP request is received at the server, it assigns all IP addresses within that range only.

With DHCP Assist enabled on a Foundry Layer 2 Switch, correct assignments are made because the Layer 2 Switch provides the stamping service.

How DHCP Assist Works

Upon initiation of a DHCP session, the client sends out a DHCP discovery packet for an address from the DHCP server as seen in Figure 29.10. When the DHCP discovery packet is received at a Foundry Layer 2 Switch with the DHCP Assist feature enabled, the gateway address configured on the receiving interface is inserted into the packet. This address insertion is also referred to as stamping.

Figure 29.10 DHCP Requests in a Network with DHCP Assist Operating on a FastIron

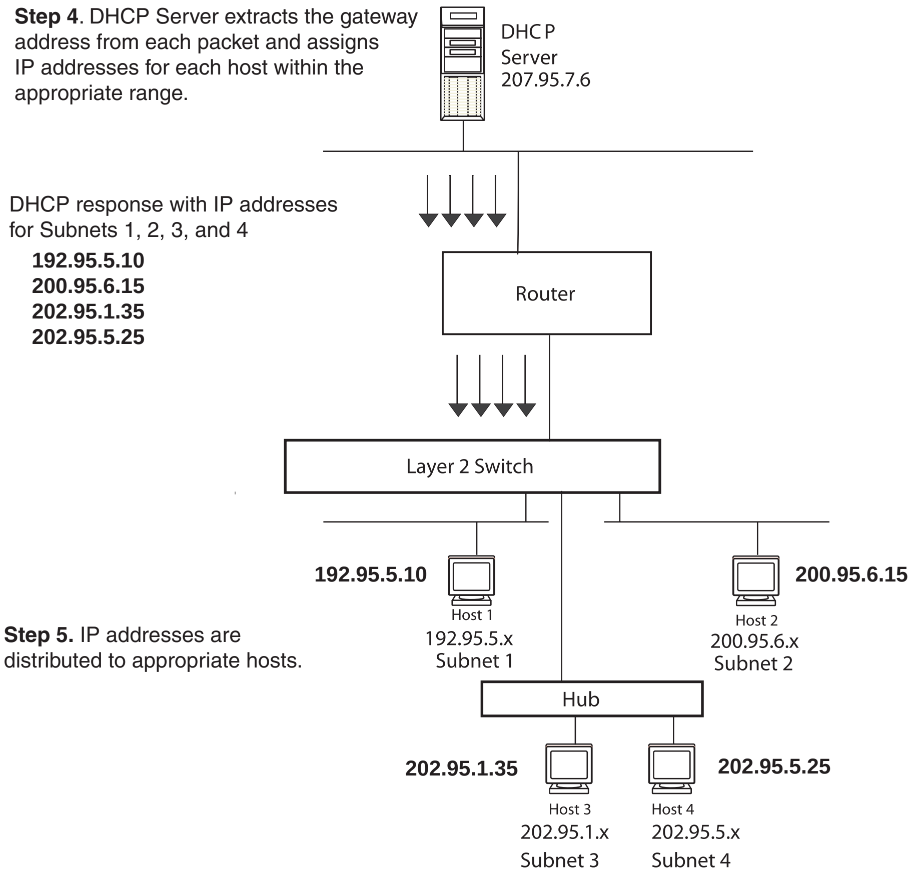

DHCP
Server
207.95.7.6
Step 3:
Router forwards the DHCP request to the server without touching the gateway address inserted in the packet by the switch
Router
Step 2:
FastIron stamps each DHCP request with the gateway address of the corresponding subnet of the receiving port.
Gateway addresses:
192.95.5.1
200.95.6.1
202.95.1.1
202.95.5.1
Layer 2 Switch
Interface 2
Interface 14
Host 1
192.95.5.x
Subnet 1
Interface 8
Host 2
200.95.6.x
Subnet 2
Step 1:
DHCP IP address requests for Hosts 1,2,3 and 4 in Subnets 1, 2, 3 and 4
Hub
Host 3
Host 4
202.95.1.x
Subnet 3
202.95.5.x
Subnet 4

When the stamped DHCP discovery packet is then received at the router, it is forwarded to the DHCP server. The DHCP server then extracts the gateway address from each request and assigns an available IP address within the corresponding IP subnet (Figure 29.11). The IP address is then forwarded back to the workstation that originated the request.

NOTE: When DHCP Assist is enabled on any port, Layer 2 broadcast packets are forwarded by the CPU. Unknown unicast and multicast packets are still forwarded in hardware, although selective packets such as IGMP, are sent to the CPU for analysis. When DHCP Assist is not enabled, Layer 2 broadcast packets are forwarded in hardware.

NOTE: The DHCP relay function of the connecting router must be turned on.

Figure 29.11 DHCP Offers are Forwarded back toward the Requestors



NOTE: When DHCP Assist is enabled on any port, Layer 2 broadcast packets are forwarded by the CPU. Unknown unicast and multicast packets are still forwarded in hardware, although selective packets such as IGMP are sent to the CPU for analysis. When DHCP Assist is not enabled, Layer 2 broadcast packets are forwarded in hardware.

Configuring DHCP Assist

You can associate a gateway list with a port. You must configure a gateway list when DHCP Assist is enabled on a Foundry Layer 2 Switch. The gateway list contains a gateway address for each subnet that will be requesting addresses from a DHCP server. The list allows the stamping process to occur. Each gateway address defined on the Layer 2 Switch corresponds to an IP address of the Foundry router interface or other router involved.

Up to eight addresses can be defined for each gateway list in support of ports that are multi-homed. When multiple IP addresses are configured for a gateway list, the Layer 2 Switch inserts the addresses into the discovery packet in a round robin fashion.

Up to 32 gateway lists can be defined for each Layer 2 Switch.

EXAMPLES:

To create the configuration indicated in Figure 29.10 and Figure 29.11:

```
FastIron(config)#dhcp-gateway-list 1 192.95.5.1
FastIron(config)#dhcp-gateway-list 2 200.95.6.1
FastIron(config)#dhcp-gateway-list 3 202.95.1.1 202.95.5.1
FastIron(config)#int e 2
FastIron(config-if-e1000-2)#dhcp-gateway-list 1
FastIron(config-if-e1000-2)#int e8
FastIron(config-if-e1000-8)#dhcp-gateway-list 3
FastIron(config-if-e1000-8)#int e 14
FastIron(config-if-e1000-14)#dhcp-gateway-list 2
```

Syntax: dhcp-gateway-list <num> <ip-addr>

Displaying IP Configuration Information and Statistics

The following sections describe IP display options for Layer 3 Switches and Layer 2 Switches.

- To display IP information on a Layer 3 Switch, see "Displaying IP Information – Layer 3 Switches" on page 29-64.
- To display IP information on a Layer 2 Switch, see "Displaying IP Information – Layer 2 Switches" on page 29-79.

Changing the Network Mask Display to Prefix Format

By default, the CLI displays network masks in classical IP address format (example: 255.255.255.0). You can change the displays to prefix format (example: /18) on a Layer 3 Switch or Layer 2 Switch using the following CLI method.

NOTE: This option does not affect how information is displayed in the Web management interface.

To enable CIDR format for displaying network masks, entering the following command at the global CONFIG level of the CLI:

```
FastIron(config)#ip show-subnet-length
```

Syntax: [no] ip show-subnet-length

Displaying IP Information – Layer 3 Switches

You can display the following IP configuration information statistics on Layer 3 Switches:

- Global IP parameter settings and IP access policies – see "Displaying Global IP Configuration Information" on page 29-65.
- CPU utilization statistics – see "Displaying CPU Utilization Statistics" on page 29-67.
- IP interfaces – see "Displaying IP Interface Information" on page 29-68.
- ARP entries – see "Displaying ARP Entries" on page 29-70.
- Static ARP entries – see "Displaying ARP Entries" on page 29-70.
- IP forwarding cache – see "Displaying the Forwarding Cache" on page 29-72.
- IP route table – see "Displaying the IP Route Table" on page 29-73.
- IP traffic statistics – see "Displaying IP Traffic Statistics" on page 29-76.

The sections below describe how to display this information.

In addition to the information described below, you can display the following IP information. This information is described in other parts of this guide.

- RIP
- OSPF
- BGP4
- DVMRP
- PIM
- VRRP or VRRPE

Displaying Global IP Configuration Information

To display IP configuration information, enter the following command at any CLI level:

```
FastIron#show ip

Global Settings
  ttl: 64, arp-age: 10, bootp-relay-max-hops: 4
  router-id : 207.95.11.128
  enabled : UDP-Broadcast-Forwarding  IRDP  Proxy-ARP  RARP  OSPF
  disabled: BGP4 Load-Sharing  RIP  DVMRP  FSRP  VRRP
Static Routes
  Index   IP Address        Subnet Mask       Next Hop Router   Metric Distance
  1       0.0.0.0           0.0.0.0           209.157.23.2      1      1
Policies
  Index   Action   Source           Destination      Protocol   Port  Operator
  1       deny     209.157.22.34    209.157.22.26    tcp        http  =
  64      permit   any              any
```

Syntax: show ip

NOTE: This command has additional options, which are explained in other sections in this guide, including the sections below this one.

This display shows the following information.

Table 29.7: CLI Display of Global IP Configuration Information – Layer 3 Switch

This Field...	Displays...
Global settings	
ttl	The Time-To-Live (TTL) for IP packets. The TTL specifies the maximum number of router hops a packet can travel before reaching the Foundry router. If the packet's TTL value is higher than the value specified in this field, the Foundry router drops the packet. To change the maximum TTL, see "Changing the TTL Threshold" on page 29-31.
arp-age	The ARP aging period. This parameter specifies how many minutes an inactive ARP entry remains in the ARP cache before the router ages out the entry. To change the ARP aging period, see "Changing the ARP Aging Period" on page 29-27.
bootp-relay-max-hops	The maximum number of hops away a BootP server can be located from the Foundry router and still be used by the router's clients for network booting. To change this value, see "Changing the Maximum Number of Hops to a BootP Relay Server" on page 29-51.
router-id	The 32-bit number that uniquely identifies the Foundry router. By default, the router ID is the numerically lowest IP interface configured on the router. To change the router ID, see "Changing the Router ID" on page 29-24.
enabled	The IP-related protocols that are enabled on the router.
disabled	The IP-related protocols that are disabled on the router.

Table 29.7: CLI Display of Global IP Configuration Information – Layer 3 Switch (Continued)

This Field...	Displays...
Static routes	
Index	The row number of this entry in the IP route table.
IP Address	The IP address of the route's destination.
Subnet Mask	The network mask for the IP address.
Next Hop Router	The IP address of the router interface to which the Foundry router sends packets for the route.
Metric	The cost of the route. Usually, the metric represents the number of hops to the destination.
Distance	The administrative distance of the route. The default administrative distance for static IP routes in Foundry routers is 1. To list the default administrative distances for all types of routes or to change the administrative distance of a static route, see "Changing Administrative Distances" on page 38-28.
Policies	
Index	The policy number. This is the number you assigned the policy when you configured it.
Action	The action the router takes if a packet matches the comparison values in the policy. The action can be one of the following: • deny – The router drops packets that match this policy. • permit – The router forwards packets that match this policy.
Source	The source IP address the policy matches.
Destination	The destination IP address the policy matches.
Protocol	The IP protocol the policy matches. The protocol can be one of the following: • ICMP • IGMP • IGRP • OSPF • TCP • UDP
Port	The Layer 4 TCP or UDP port the policy checks for in packets. The port can be displayed by its number or, for port types the router recognizes, by the well-known name. For example, TCP port 80 can be displayed as HTTP. **Note**: This field applies only if the IP protocol is TCP or UDP.
Operator	The comparison operator for TCP or UDP port names or numbers. **Note**: This field applies only if the IP protocol is TCP or UDP.

Displaying CPU Utilization Statistics

You can display CPU utilization statistics for IP protocols using the **show process cpu** command.

The **show process cpu** command includes CPU utilization statistics for ACL, 802.1x, and L2VLAN. L2VLAN contains any packet transmitted to a VLAN by the CPU, including unknown unicast, multicast, broadcast, and CPU forwarded Layer 2 traffic.

To display CPU utilization statistics for the previous one-second, one-minute, five-minute, and fifteen-minute intervals, enter the following command at any level of the CLI:

```
FastIron#show process cpu
Process Name   5Sec(%)   1Min(%)   5Min(%)   15Min(%)   Runtime(ms)
ACL              0.00      0.00      0.00      0.00                0
ARP              0.01      0.01      0.01      0.01              714
BGP              0.00      0.00      0.00      0.00                0
DOT1X            0.00      0.00      0.00      0.00                0
GVRP             0.00      0.00      0.00      0.00                0
ICMP             0.00      0.00      0.00      0.00              161
IP               0.00      0.00      0.00      0.00              229
L2VLAN           0.01      0.00      0.00      0.01              673
OSPF             0.00      0.00      0.00      0.00                0
RIP              0.00      0.00      0.00      0.00                9
STP              0.00      0.00      0.00      0.00                7
VRRP             0.00      0.00      0.00      0.00                0
```

If the software has been running less than 15 minutes (the maximum interval for utilization statistics), the command indicates how long the software has been running. Here is an example:

```
FastIron#show process cpu
The system has only been up for 6 seconds.
Process Name   5Sec(%)   1Min(%)   5Min(%)   15Min(%)   Runtime(ms)
ACL              0.00      0.00      0.00      0.00                0
ARP              0.01      0.01      0.01      0.01              714
BGP              0.00      0.00      0.00      0.00                0
DOT1X            0.00      0.00      0.00      0.00                0
GVRP             0.00      0.00      0.00      0.00                0
ICMP             0.00      0.00      0.00      0.00              161
IP               0.00      0.00      0.00      0.00              229
L2VLAN           0.01      0.00      0.00      0.01              673
OSPF             0.00      0.00      0.00      0.00                0
RIP              0.00      0.00      0.00      0.00                9
STP              0.00      0.00      0.00      0.00                7
VRRP             0.00      0.00      0.00      0.00                0
```

To display utilization statistics for a specific number of seconds, enter a command such as the following:

```
FastIron#show process cpu 2
Statistics for last 1 sec and 80 ms
Process Name   Sec(%)   Time(ms)
ACL                0      0.00
ARP                1      0.01
BGP                0      0.00
DOT1X              0      0.00
GVRP               0      0.00
ICMP               0      0.00
IP                 0      0.00
L2VLAN             1      0.01
OSPF               0      0.00
RIP                0      0.00
STP                0      0.00
VRRP               0      0.00
```

When you specify how many seconds' worth of statistics you want to display, the software selects the sample that most closely matches the number of seconds you specified. In this example, statistics are requested for the previous two seconds. The closest sample available is actually for the previous 1 second plus 80 milliseconds.

Syntax: show process cpu [<num>]

The <num> parameter specifies the number of seconds and can be from 1 – 900. If you use this parameter, the command lists the usage statistics only for the specified number of seconds. If you do not use this parameter, the command lists the usage statistics for the previous one-second, one-minute, five-minute, and fifteen-minute intervals.

Displaying IP Interface Information

To display IP interface information, enter the following command at any CLI level:

```
FastIron#show ip interface

Interface       IP-Address      OK?  Method    Status                 Protocol
Ethernet 1/1    207.95.6.173    YES  NVRAM     up                     up
Ethernet 1/2    3.3.3.3         YES  manual    up                     up
Loopback 1      1.2.3.4         YES  NVRAM     down                   down
```

Syntax: show ip interface [ethernet [<slotnum>/]<portnum>] | [loopback <num>] | [ve <num>]

This display shows the following information.

Table 29.8: CLI Display of Interface IP Configuration Information

This Field...	Displays...
Interface	The type and the slot and port number of the interface.
IP-Address	The IP address of the interface. **Note**: If an "s" is listed following the address, this is a secondary address. When the address was configured, the interface already had an IP address in the same subnet, so the software required the "secondary" option before the software could add the interface.
OK?	Whether the IP address has been configured on the interface.

Table 29.8: CLI Display of Interface IP Configuration Information (Continued)

This Field...	Displays...
Method	Whether the IP address has been saved in NVRAM. If you have set the IP address for the interface in the CLI or Web Management interface, but have not saved the configuration, the entry for the interface in the Method field is “manual”.
Status	The link status of the interface. If you have disabled the interface with the **disable** command, the entry in the Status field will be “administratively down”. Otherwise, the entry in the Status field will be either “up” or “down”.
Protocol	Whether the interface can provide two-way communication. If the IP address is configured, and the link status of the interface is up, the entry in the protocol field will be “up”. Otherwise the entry in the protocol field will be “down”.

To display detailed IP information for a specific interface, enter a command such as the following:

```
FastIron#show ip interface ethernet 1/1
Interface Ethernet 1/1
  port state: UP
  ip address: 192.168.9.51      subnet mask: 255.255.255.0
  encapsulation: ETHERNET, mtu: 1500, metric: 1
  directed-broadcast-forwarding: disabled
  proxy-arp: disabled
  ip arp-age:  10 minutes
  Ip Flow switching is disabled
  No Helper Addresses are configured.
  No inbound ip access-list is set
  No outgoing ip access-list is set
```

Displaying Interface Names in Syslog

By default an interface’s slot number (if applicable) and port number are displayed when you display Syslog messages. Beginning with release 02.5.00, you can display the name of the interface instead of its number by entering a command such as the following:

```
FastIron(config)#ip show-portname
```

This command is applied globally to all interfaces on Layer 2 Switches and Layer 3 Switches.

NOTE: This command is not support on the FGS.

Syntax: [no] Ip show-portname

When you display the messages in the Syslog, you see the interface name under the Dynamic Log Buffer section. The actual interface number is appended to the interface name. For example, if the interface name is "lab" and its port number is "2", you see "lab2" displayed as in the example below:

```
FastIron>#show logging
Syslog logging: enabled (0 messages dropped, 0 flushes, 0 overruns)
    Buffer logging: level ACDMEINW, 3 messages logged
    level code: A=alert C=critical D=debugging M=emergency E=error
         I=informational N=notification W=warning
Static Log Buffer:
Dec 15 19:04:14:A:Fan 1, fan on right connector, failed

Dynamic Log Buffer (50 entries):
Dec 15 18:46:17:I:Interface ethernet Lab2, state up
Dec 15 18:45:15:I:Warm start
```

Displaying ARP Entries

You can display the ARP cache and the static ARP table. The ARP cache contains entries for devices attached to the Layer 3 Switch. The static ARP table contains the user-configured ARP entries. An entry in the static ARP table enters the ARP cache when the entry's interface comes up.

The tables require separate display commands or Web management options.

Displaying the ARP Cache

To display the contents of the ARP cache, enter the following command at any CLI level:

```
FastIron#show arp

Total number of ARP entries: 5
     IP Address          MAC Address          Type        Age     Port
1     207.95.6.102        0800.5afc.ea21       Dynamic     0          6
2     207.95.6.18         00a0.24d2.04ed       Dynamic     3          6
3     207.95.6.54         00a0.24ab.cd2b       Dynamic     0          6
4     207.95.6.101        0800.207c.a7fa       Dynamic     0          6
5     207.95.6.211        00c0.2638.ac9c       Dynamic     0          6
```

Syntax: show arp [ethernet [<slotnum>/]<portnum> | mac-address <xxxx.xxxx.xxxx> [<mask>] | <ip-addr> [<ip-mask>]] [<num>]

The <slotnum> parameter is required on chassis devices.

The <portnum> parameter lets you restrict the display to entries for a specific port.

The **mac-address** <xxxx.xxxx.xxxx> parameter lets you restrict the display to entries for a specific MAC address.

The <mask> parameter lets you specify a mask for the **mac-address** <xxxx.xxxx.xxxx> parameter, to display entries for multiple MAC addresses. Specify the MAC address mask as "f"s and "0"s, where "f"s are significant bits.

The <ip-addr> and <ip-mask> parameters let you restrict the display to entries for a specific IP address and network mask. Specify the IP address masks in standard decimal mask format (for example, 255.255.0.0).

NOTE: The <ip-mask> parameter and <mask> parameter perform different operations. The <ip-mask> parameter specifies the network mask for a specific IP address, whereas the <mask> parameter provides a filter for displaying multiple MAC addresses that have specific values in common.

The <num> parameter lets you display the table beginning with a specific entry number.

NOTE: The entry numbers in the ARP cache are not related to the entry numbers for static ARP table entries.

This display shows the following information. The number in the left column of the CLI display is the row number of the entry in the ARP cache. This number is not related to the number you assign to static MAC entries in the static ARP table.

Table 29.9: CLI Display of ARP Cache

This Field...	Displays...
IP Address	The IP address of the device.
MAC Address	The MAC address of the device.
Type	The type, which can be one of the following: • Dynamic – The Layer 3 Switch learned the entry from an incoming packet. • Static – The Layer 3 Switch loaded the entry from the static ARP table when the device for the entry was connected to the Layer 3 Switch.
Age	The number of minutes the entry has remained unused. If this value reaches the ARP aging period, the entry is removed from the table. To display the ARP aging period, see “Displaying Global IP Configuration Information” on page 29-65. To change the ARP aging interval, see “Changing the ARP Aging Period” on page 29-27. **Note**: Static entries do not age out.
Port	The port on which the entry was learned.

Displaying the Static ARP Table

To display the static ARP table instead of the ARP cache, enter the following command at any CLI level:

```
FastIron#show ip static-arp

Static ARP table size: 512, configurable from 512 to 1024
  Index   IP Address          MAC Address         Port
  1       207.95.6.111        0800.093b.d210      1/1
  3       207.95.6.123        0800.093b.d211      1/1
```

This example shows two static entries. Note that since you specify an entry’s index number when you create the entry, it is possible for the range of index numbers to have gaps, as shown in this example.

NOTE: The entry number you assign to a static ARP entry is not related to the entry numbers in the ARP cache.

Syntax: show ip static-arp [ethernet [<slotnum>/]<portnum> | mac-address <xxxx.xxxx.xxxx> [<mask>] | <ip-addr> [<ip-mask>]] [<num>]

The <slotnum> parameter is required on chassis devices.

The <portnum> parameter lets you restrict the display to entries for a specific port.

The **mac-address** <xxxx.xxxx.xxxx> parameter lets you restrict the display to entries for a specific MAC address.

The <mask> parameter lets you specify a mask for the **mac-address** <xxxx.xxxx.xxxx> parameter, to display entries for multiple MAC addresses. Specify the MAC address mask as “f”s and “0”s, where “f”s are significant bits.

The <ip-addr> and <ip-mask> parameters let you restrict the display to entries for a specific IP address and network mask. Specify the IP address masks in standard decimal mask format (for example, 255.255.0.0).

NOTE: The <ip-mask> parameter and <mask> parameter perform different operations. The <ip-mask> parameter specifies the network mask for a specific IP address, whereas the <mask> parameter provides a filter for displaying multiple MAC addresses that have specific values in common.

The <num> parameter lets you display the table beginning with a specific entry number.

Table 29.10: CLI Display of Static ARP Table

This Field...	Displays...
Static ARP table size	The maximum number of static entries that can be configured on the device using the current memory allocation. The range of valid memory allocations for static ARP entries is listed after the current allocation. To change the memory allocation for static ARP entries, see “Changing the Maximum Number of Entries the Static ARP Table Can Hold” on page 29-30.
Index	The number of this entry in the table. You specify the entry number when you create the entry.
IP Address	The IP address of the device.
MAC Address	The MAC address of the device.
Port	The port attached to the device the entry is for.

Displaying the Forwarding Cache

To display the IP forwarding cache, enter the following command at any CLI level:

```
FastIron#show ip cache

Total number of cache entries: 3
D:Dynamic  P:Permanent  F:Forward  U:Us  C:Complex Filter
W:Wait ARP  I:ICMP Deny  K:Drop  R:Fragment  S:Snap Encap
      IP Address       Next Hop         MAC              Type  Port  Vlan  Pri
1     192.168.1.11     DIRECT           0000.0000.0000   PU    n/a         0
2     192.168.1.255    DIRECT           0000.0000.0000   PU    n/a         0
3     255.255.255.255  DIRECT           0000.0000.0000   PU    n/a         0
```

Syntax: show ip cache [<ip-addr>] | [<num>]

The <ip-addr> parameter displays the cache entry for the specified IP address.

The <num> parameter displays the cache beginning with the row following the number you enter. For example, to begin displaying the cache at row 10, enter the following command: **show ip cache 9**.

The **show ip cache** command displays the following information.

Table 29.11: CLI Display of IP Forwarding Cache – Layer 3 Switch

This Field...	Displays...
IP Address	The IP address of the destination.

Table 29.11: CLI Display of IP Forwarding Cache – Layer 3 Switch (Continued)

This Field...	Displays...
Next Hop	The IP address of the next-hop router to the destination. This field contains either an IP address or the value DIRECT. DIRECT means the destination is either directly attached or the destination is an address on this Foundry device. For example, the next hop for loopback addresses and broadcast addresses is shown as DIRECT.
MAC	The MAC address of the destination. **Note**: If the entry is type U (indicating that the destination is this Foundry device), the address consists of zeroes.
Type	The type of host entry, which can be one or more of the following: • D – Dynamic • P – Permanent • F – Forward • U – Us • C – Complex Filter • W – Wait ARP • I – ICMP Deny • K – Drop • R – Fragment • S – Snap Encap
Port	The port through which this device reaches the destination. For destinations that are located on this device, the port number is shown as “n/a”.
VLAN	Indicates the VLAN(s) the listed port is in.
Pri	The QoS priority of the port or VLAN.

Displaying the IP Route Table

To display the IP route table, enter the following command at any CLI level:

```
FastIron#show ip route

Total number of IP routes: 514
Start index: 1  B:BGP D:Connected  R:RIP  S:Static  O:OSPF *:Candidate default

Destination       NetMask           Gateway           Port   Cost   Type
1.1.0.0           255.255.0.0       99.1.1.2          1/1    2      R
1.2.0.0           255.255.0.0       99.1.1.2          1/1    2      R
1.3.0.0           255.255.0.0       99.1.1.2          1/1    2      R
1.4.0.0           255.255.0.0       99.1.1.2          1/1    2      R
1.5.0.0           255.255.0.0       99.1.1.2          1/1    2      R
1.6.0.0           255.255.0.0       99.1.1.2          1/1    2      R
1.7.0.0           255.255.0.0       99.1.1.2          1/1    2      R
1.8.0.0           255.255.0.0       99.1.1.2          1/1    2      R
1.9.0.0           255.255.0.0       99.1.1.2          1/1    2      R
1.10.0.0          255.255.0.0       99.1.1.2          1/1    2      S
```

Syntax: show ip route [<ip-addr> [<ip-mask>] [longer] [none-bgp]] | <num> | bgp | direct | ospf | rip | static

The <ip-addr> parameter displays the route to the specified IP address.

The <ip-mask> parameter lets you specify a network mask or, if you prefer CIDR format, the number of bits in the network mask. If you use CIDR format, enter a forward slash immediately after the IP address, then enter the number of mask bits (for example: 209.157.22.0/24 for 209.157.22.0 255.255.255.0).

The **longer** parameter applies only when you specify an IP address and mask. This option displays only the routes for the specified IP address and mask. See the example below.

The **none-bgp** parameter displays only the routes that did not come from BGP4.

The <num> option display the route table entry whose row number corresponds to the number you specify. For example, if you want to display the tenth row in the table, enter "10".

The **bgp** option displays the BGP4 routes.

The **direct** option displays only the IP routes that are directly attached to the Layer 3 Switch.

The **ospf** option displays the OSPF routes.

The **rip** option displays the RIP routes.

The **static** option displays only the static IP routes.

The default routes are displayed first.

Here is an example of how to use the **direct** option. To display only the IP routes that go to devices directly attached to the Layer 3 Switch:

```
FastIron#show ip route direct
Start index: 1  B:BGP D:Connected  R:RIP  S:Static  O:OSPF *:Candidate default

      Destination       NetMask           Gateway           Port   Cost   Type
      209.157.22.0      255.255.255.0     0.0.0.0           4/11   1      D
```

Notice that the route displayed in this example has "D" in the Type field, indicating the route is to a directly connected device.

Here is an example of how to use the **static** option. To display only the static IP routes:

```
FastIron#show ip route static
Start index: 1  B:BGP D:Connected  R:RIP  S:Static  O:OSPF *:Candidate default

      Destination       NetMask           Gateway           Port   Cost   Type
      192.144.33.11     255.255.255.0     209.157.22.12     1/1    2      S
```

Notice that the route displayed in this example has "S" in the Type field, indicating the route is static.

Here is an example of how to use the **longer** option. To display only the routes for a specified IP address and mask, enter a command such as the following:

```
FastIron#show ip route 209.159.0.0/16 longer

Starting index: 1 B:BGP D:Directly-Connected R:RIP S:Static O:OSPF
Destination NetMask Gateway Port Cost Type

52 209.159.38.0 255.255.255.0 207.95.6.101 1/1 1 S
53 209.159.39.0 255.255.255.0 207.95.6.101 1/1 1 S
54 209.159.40.0 255.255.255.0 207.95.6.101 1/1 1 S
55 209.159.41.0 255.255.255.0 207.95.6.101 1/1 1 S
56 209.159.42.0 255.255.255.0 207.95.6.101 1/1 1 S
57 209.159.43.0 255.255.255.0 207.95.6.101 1/1 1 S
58 209.159.44.0 255.255.255.0 207.95.6.101 1/1 1 S
59 209.159.45.0 255.255.255.0 207.95.6.101 1/1 1 S
60 209.159.46.0 255.255.255.0 207.95.6.101 1/1 1 S
```

This example shows all the routes for networks beginning with 209.159. The mask value and **longer** parameter specify the range of network addresses to be displayed. In this example, all routes within the range 209.159.0.0 – 209.159.255.255 are listed.

The **summary** option displays a summary of the information in the IP route table. The following is an example of the output from this command:

EXAMPLES:

```
FastIron#show ip route summary

IP Routing Table - 35 entries:
  6 connected, 28 static, 0 RIP, 1 OSPF, 0 BGP, 0 ISIS, 0 MPLS
  Number of prefixes:
  /0: 1 /16: 27 /22: 1 /24: 5 /32: 1
```

Syntax: show ip route summary

In this example, the IP route table contains 35 entries. Of these entries, 6 are directly connected devices, 28 are static routes, and 1 route was calculated through OSPF. One of the routes has a zero-bit mask (this is the default route), 27 have a 22-bit mask, 5 have a 24-bit mask, and 1 has a 32-bit mask.

The following table lists the information displayed by the **show ip route** command.

Table 29.12: CLI Display of IP Route Table

This Field...	Displays...
Destination	The destination network of the route.
NetMask	The network mask of the destination address.
Gateway	The next-hop router.
Port	The port through which this router sends packets to reach the route's destination.
Cost	The route's cost.
Type	The route type, which can be one of the following: • B – The route was learned from BGP. • D – The destination is directly connected to this Layer 3 Switch. • R – The route was learned from RIP. • S – The route is a static route. • * – The route is a candidate default route. • O – The route is an OSPF route. Unless you use the **ospf** option to display the route table, “O” is used for all OSPF routes. If you do use the **ospf** option, the following type codes are used: • O – OSPF intra area route (within the same area). • IA – The route is an OSPF inter area route (a route that passes from one area into another). • E1 – The route is an OSPF external type 1 route. • E2 – The route is an OSPF external type 2 route.

Clearing IP Routes

If needed, you can clear the entire route table or specific individual routes.

To clear all routes from the IP route table:

```
FastIron#clear ip route
```

To clear route 209.157.22.0/24 from the IP routing table:

```
FastIron#clear ip route 209.157.22.0/24
```

Syntax: clear ip route [<ip-addr> <ip-mask>]

or

Syntax: clear ip route [<ip-addr>/<mask-bits>]

Displaying IP Traffic Statistics

To display IP traffic statistics, enter the following command at any CLI level:

```
FastIron#show ip traffic

IP Statistics

  139 received, 145 sent, 0 forwarded
  0 filtered, 0 fragmented, 0 reassembled, 0 bad header
  0 no route, 0 unknown proto, 0 no buffer, 0 other errors

ICMP Statistics
Received:
  0 total, 0 errors, 0 unreachable, 0 time exceed
  0 parameter, 0 source quench, 0 redirect, 0 echo,
  0 echo reply, 0 timestamp, 0 timestamp reply, 0 addr mask
  0 addr mask reply, 0 irdp advertisement, 0 irdp solicitation
Sent:
  0 total, 0 errors, 0 unreachable, 0 time exceed
  0 parameter, 0 source quench, 0 redirect, 0 echo,
  0 echo reply, 0 timestamp, 0 timestamp reply, 0 addr mask
  0 addr mask reply, 0 irdp advertisement, 0 irdp solicitation

UDP Statistics
  1 received, 0 sent, 1 no port, 0 input errors

TCP Statistics
  0 active opens, 0 passive opens, 0 failed attempts
  0 active resets, 0 passive resets, 0 input errors
  138 in segments, 141 out segments, 4 retransmission

RIP Statistics
  0 requests sent, 0 requests received
  0 responses sent, 0 responses received
  0 unrecognized, 0 bad version, 0 bad addr family, 0 bad req format
  0 bad metrics, 0 bad resp format, 0 resp not from rip port
  0 resp from loopback, 0 packets rejected
```

The **show ip traffic** command displays the following information.

Table 29.13: CLI Display of IP Traffic Statistics – Layer 3 Switch

This Field...	Displays...
IP statistics	
received	The total number of IP packets received by the device.
sent	The total number of IP packets originated and sent by the device.
forwarded	The total number of IP packets received by the device and forwarded to other devices.
filtered	The total number of IP packets filtered by the device.
fragmented	The total number of IP packets fragmented by this device to accommodate the MTU of this device or of another device.
reassembled	The total number of fragmented IP packets that this device re-assembled.
bad header	The number of IP packets dropped by the device due to a bad packet header.
no route	The number of packets dropped by the device because there was no route.
unknown proto	The number of packets dropped by the device because the value in the Protocol field of the packet header is unrecognized by this device.
no buffer	This information is used by Foundry customer support.
other errors	The number of packets dropped due to error types other than those listed above.
ICMP statistics The ICMP statistics are derived from RFC 792, “Internet Control Message Protocol”, RFC 950, “Internet Standard Subnetting Procedure”, and RFC 1256, “ICMP Router Discovery Messages”. Statistics are organized into Sent and Received. The field descriptions below apply to each.	
total	The total number of ICMP messages sent or received by the device.
errors	This information is used by Foundry customer support.
unreachable	The number of Destination Unreachable messages sent or received by the device.
time exceed	The number of Time Exceeded messages sent or received by the device.
parameter	The number of Parameter Problem messages sent or received by the device.
source quench	The number of Source Quench messages sent or received by the device.
redirect	The number of Redirect messages sent or received by the device.
echo	The number of Echo messages sent or received by the device.
echo reply	The number of Echo Reply messages sent or received by the device.
timestamp	The number of Timestamp messages sent or received by the device.
timestamp reply	The number of Timestamp Reply messages sent or received by the device.
addr mask	The number of Address Mask Request messages sent or received by the device.
addr mask reply	The number of Address Mask Replies messages sent or received by the device.

Table 29.13: CLI Display of IP Traffic Statistics – Layer 3 Switch (Continued)

This Field...	Displays...
irdp advertisement	The number of ICMP Router Discovery Protocol (IRDP) Advertisement messages sent or received by the device.
irdp solicitation	The number of IRDP Solicitation messages sent or received by the device.
UDP statistics	
received	The number of UDP packets received by the device.
sent	The number of UDP packets sent by the device.
no port	The number of UDP packets dropped because they did not have a valid UDP port number.
input errors	This information is used by Foundry customer support.
TCP statistics The TCP statistics are derived from RFC 793, “Transmission Control Protocol”.	
active opens	The number of TCP connections opened by sending a TCP SYN to another device.
passive opens	The number of TCP connections opened by this device in response to connection requests (TCP SYNs) received from other devices.
failed attempts	This information is used by Foundry customer support.
active resets	The number of TCP connections this device reset by sending a TCP RESET message to the device at the other end of the connection.
passive resets	The number of TCP connections this device reset because the device at the other end of the connection sent a TCP RESET message.
input errors	This information is used by Foundry customer support.
in segments	The number of TCP segments received by the device.
out segments	The number of TCP segments sent by the device.
retransmission	The number of segments that this device retransmitted because the retransmission timer for the segment had expired before the device at the other end of the connection had acknowledged receipt of the segment.
RIP statistics The RIP statistics are derived from RFC 1058, “Routing Information Protocol”.	
requests sent	The number of requests this device has sent to another RIP router for all or part of its RIP routing table.
requests received	The number of requests this device has received from another RIP router for all or part of this device’s RIP routing table.
responses sent	The number of responses this device has sent to another RIP router’s request for all or part of this device’s RIP routing table.
responses received	The number of responses this device has received to requests for all or part of another RIP router’s routing table.
unrecognized	This information is used by Foundry customer support.

Table 29.13: CLI Display of IP Traffic Statistics – Layer 3 Switch (Continued)

This Field...	Displays...
bad version	The number of RIP packets dropped by the device because the RIP version was either invalid or is not supported by this device.
bad addr family	The number of RIP packets dropped because the value in the Address Family Identifier field of the packet's header was invalid.
bad req format	The number of RIP request packets this router dropped because the format was bad.
bad metrics	This information is used by Foundry customer support.
bad resp format	The number of responses to RIP request packets dropped because the format was bad.
resp not from rip port	This information is used by Foundry customer support.
resp from loopback	The number of RIP responses received from loopback interfaces.
packets rejected	This information is used by Foundry customer support.

Displaying IP Information – Layer 2 Switches

You can display the following IP configuration information statistics on Layer 2 Switches:

- Global IP settings – see "Displaying Global IP Configuration Information" on page 29-79.
- ARP entries – see "Displaying ARP Entries" on page 29-80.
- IP traffic statistics – see "Displaying IP Traffic Statistics" on page 29-81.

Displaying Global IP Configuration Information

To display the Layer 2 Switch's IP address and default gateway, enter the following command:

```
FastIron#show ip

     Switch IP address: 192.168.1.2

           Subnet mask: 255.255.255.0

Default router address: 192.168.1.1
   TFTP server address: None
Configuration filename: None
        Image filename: None
```

Syntax: show ip

This display shows the following information.

Table 29.14: CLI Display of Global IP Configuration Information – Layer 2 Switch

This Field...	Displays...
IP configuration	
Switch IP address	The management IP address configured on the Layer 2 Switch. Specify this address for Telnet or Web management access.
Subnet mask	The subnet mask for the management IP address.

Table 29.14: CLI Display of Global IP Configuration Information – Layer 2 Switch

This Field...	Displays...
Default router address	The address of the default gateway, if you specified one.
Most recent TFTP access	
TFTP server address	The IP address of the most-recently contacted TFTP server, if the switch has contacted a TFTP server since the last time the software was reloaded or the switch was rebooted.
Configuration filename	The name under which the Layer 2 Switch's startup-config file was uploaded or downloaded during the most recent TFTP access.
Image filename	The name of the Layer 2 Switch flash image (system software file) that was uploaded or downloaded during the most recent TFTP access.

Displaying ARP Entries

To display the entries the Layer 2 Switch has placed in its ARP cache, enter the following command from any level of the CLI:

```
FastIron#show arp

         IP               Mac        Port Age VlanId
192.168.1.170       0010.5a11.d042     7   0      1
Total Arp Entries : 1
```

Syntax: show arp

This display shows the following information.

Table 29.15: CLI Display of ARP Cache

This Field...	Displays...
IP	The IP address of the device.
Mac	The MAC address of the device. **Note**: If the MAC address is all zeros, the entry is for the default gateway, but the Layer 2 Switch does not have a link to the gateway.
Port	The port on which the entry was learned.
Age	The number of minutes the entry has remained unused. If this value reaches the ARP aging period, the entry is removed from the cache.
VlanId	The VLAN the port that learned the entry is in. **Note**: If the MAC address is all zeros, this field shows a random VLAN ID, since the Layer 2 Switch does not yet know which port the device for this entry is attached to.
Total ARP Entries	The number of entries in the ARP cache.

Displaying IP Traffic Statistics

To display IP traffic statistics on a Layer 2 Switch, enter the following command at any CLI level:

```
FastIron#show ip traffic

IP Statistics
  27 received, 24 sent
  0 fragmented, 0 reassembled, 0 bad header
  0 no route, 0 unknown proto, 0 no buffer, 0 other errors

ICMP Statistics
Received:
  0 total, 0 errors, 0 unreachable, 0 time exceed
  0 parameter, 0 source quench, 0 redirect, 0 echo,
  0 echo reply, 0 timestamp, 0 timestamp rely, 0 addr mask
  0 addr mask reply, 0 irdp advertisement, 0 irdp solicitation
Sent:
  0 total, 0 errors, 0 unreachable, 0 time exceed
  0 parameter, 0 source quench, 0 redirect, 0 echo,
  0 echo reply, 0 timestamp, 0 timestamp rely, 0 addr mask
  0 addr mask reply, 0 irdp advertisement, 0 irdp solicitation

UDP Statistics
  0 received, 0 sent, 0 no port, 0 input errors

TCP Statistics
  1 current active tcbs, 4 tcbs allocated, 0 tcbs freed 0 tcbs protected
  0 active opens, 0 passive opens, 0 failed attempts
  0 active resets, 0 passive resets, 0 input errors
 27 in segments, 24 out segments, 0 retransmission
```

Syntax: show ip traffic

The **show ip traffic** command displays the following information.

Table 29.16: CLI Display of IP Traffic Statistics – Layer 2 Switch

This Field...	Displays...
IP statistics	
received	The total number of IP packets received by the device.
sent	The total number of IP packets originated and sent by the device.
fragmented	The total number of IP packets fragmented by this device to accommodate the MTU of this device or of another device.
reassembled	The total number of fragmented IP packets that this device re-assembled.
bad header	The number of IP packets dropped by the device due to a bad packet header.
no route	The number of packets dropped by the device because there was no route.
unknown proto	The number of packets dropped by the device because the value in the Protocol field of the packet header is unrecognized by this device.
no buffer	This information is used by Foundry customer support.

Table 29.16: CLI Display of IP Traffic Statistics – Layer 2 Switch (Continued)

This Field...	Displays...
other errors	The number of packets that this device dropped due to error types other than the types listed above.
ICMP statistics The ICMP statistics are derived from RFC 792, “Internet Control Message Protocol”, RFC 950, “Internet Standard Subnetting Procedure”, and RFC 1256, “ICMP Router Discovery Messages”. Statistics are organized into Sent and Received. The field descriptions below apply to each.	
total	The total number of ICMP messages sent or received by the device.
errors	This information is used by Foundry customer support.
unreachable	The number of Destination Unreachable messages sent or received by the device.
time exceed	The number of Time Exceeded messages sent or received by the device.
parameter	The number of Parameter Problem messages sent or received by the device.
source quench	The number of Source Quench messages sent or received by the device.
redirect	The number of Redirect messages sent or received by the device.
echo	The number of Echo messages sent or received by the device.
echo reply	The number of Echo Reply messages sent or received by the device.
timestamp	The number of Timestamp messages sent or received by the device.
timestamp reply	The number of Timestamp Reply messages sent or received by the device.
addr mask	The number of Address Mask Request messages sent or received by the device.
addr mask reply	The number of Address Mask Replies messages sent or received by the device.
irdp advertisement	The number of ICMP Router Discovery Protocol (IRDP) Advertisement messages sent or received by the device.
irdp solicitation	The number of IRDP Solicitation messages sent or received by the device.
UDP statistics	
received	The number of UDP packets received by the device.
sent	The number of UDP packets sent by the device.
no port	The number of UDP packets dropped because the packet did not contain a valid UDP port number.
input errors	This information is used by Foundry customer support.
TCP statistics The TCP statistics are derived from RFC 793, “Transmission Control Protocol”.	
current active tcbs	The number of TCP Control Blocks (TCBs) that are currently active.
tcbs allocated	The number of TCBs that have been allocated.
tcbs freed	The number of TCBs that have been freed.
tcbs protected	This information is used by Foundry customer support.

Table 29.16: CLI Display of IP Traffic Statistics – Layer 2 Switch (Continued)

This Field...	Displays...
active opens	The number of TCP connections opened by this device by sending a TCP SYN to another device.
passive opens	The number of TCP connections opened by this device in response to connection requests (TCP SYNs) received from other devices.
failed attempts	This information is used by Foundry customer support.
active resets	The number of TCP connections this device reset by sending a TCP RESET message to the device at the other end of the connection.
passive resets	The number of TCP connections this device reset because the device at the other end of the connection sent a TCP RESET message.
input errors	This information is used by Foundry customer support.
in segments	The number of TCP segments received by the device.
out segments	The number of TCP segments sent by the device.
retransmission	The number of segments that this device retransmitted because the retransmission timer for the segment had expired before the device at the other end of the connection had acknowledged receipt of the segment.

Chapter 30

Configuring IP Multicast Protocols

This chapter describes how to configure Foundry Layer 3 Switches for Protocol Independent Multicast (PIM) and Distance Vector Multicast Routing Protocol (DVMRP). Foundry Layer 3 Switches support the following IP multicast versions:

- Internet Group Management Protocol (IGMP) V1 and V2
- Internet Group Management Protocol (IGMP) V3
- PIM Dense mode (PIM DM) V1 (draft-ietf-pim-dm-05) and V2 (draft-ietf-pim-v2-dm-03)
- PIM Sparse mode (PIM SM) V2 (RFC 2362)
- DVMRP V2 (RFC 1075)

NOTE: Each multicast protocol uses IGMP. IGMP is automatically enabled on an interface when you configure PIM or DVMRP and is disabled on the interface if you disable PIM or DVMRP.

NOTE: This chapter applies only to IP multicast routing. To configure Layer 2 IP multicast features, see "Configuring IP Multicast Traffic Reduction for the FastIron X Series Switch" on page 27-1.

Overview of IP Multicasting

Multicast protocols allow a group or channel to be accessed over different networks by multiple stations (clients) for the receipt and transmit of multicast data.

Distribution of stock quotes, video transmissions such as news services and remote classrooms, and video conferencing are all examples of applications that use multicast routing.

Foundry Layer 3 Switches support two different multicast routing protocols—Distance Vector Multicast Routing Protocol (DVMRP) and Protocol-Independent Multicast (PIM) protocol along with the Internet Group Membership Protocol (IGMP).

PIM and DVMRP are broadcast and pruning multicast protocols that deliver IP multicast datagrams. The protocols employ reverse path lookup check and pruning to allow source-specific multicast delivery trees to reach all group members. DVMRP and PIM build a different multicast tree for each source and destination host group.

NOTE: Both DVMRP and PIM can concurrently operate on different ports of a Foundry Layer 3 Switch.

Multicast Terms

The following are commonly used terms in discussing multicast-capable routers. These terms are used throughout this chapter:

Node: Refers to a router or Layer 3 Switch.

Root Node: The node that initiates the tree building process. It is also the router that sends the multicast packets down the multicast delivery tree.

Upstream: Represents the direction from which a router receives multicast data packets. An ***upstream router*** is a node that sends multicast packets.

Downstream: Represents the direction to which a router forwards multicast data packets. A ***downstream router*** is a node that receives multicast packets from upstream transmissions.

Group Presence: Means that a multicast group has been learned from one of the directly connected interfaces. Members of the multicast group are present on the router.

Intermediate nodes: Routers that are in the path between source routers and leaf routers.

Leaf nodes: Routers that do not have any downstream routers.

Multicast Tree: A unique tree is built for each source group (S,G) pair. A multicast tree is comprised of a root node and one or more nodes that are leaf or intermediate nodes.

Changing Global IP Multicast Parameters

The following configurable parameters apply to PIM-DM, PIM-SM, and DVMRP.

- Maximum number of PIM or DVMRP groups – You can change the maximum number of groups of each type for which the software will allocate memory. By default, Layer 3 Switches support up to 1024 PIM groups and 1024 DVMRP groups.
- Internet Group Membership Protocol (IGMP) V1 and V2 parameters – You can change the query interval, group membership time, and maximum response time.
- Hardware forwarding of fragmented IP multicast packets – You can enable the Layer 3 Switch to forward all fragments of fragmented IP multicast packets in hardware.

Changing Dynamic Memory Allocation for IP Multicast Groups

Layer 3 Switches support up to 1024 PIM groups and 1024 DVMRP groups by default. Memory for the groups is allocated dynamically as needed. For each protocol, previous releases support a maximum of 255 groups and 255 IGMP memberships.

NOTE: The number of interface groups you can configure for DVMRP and PIM is unlimited; therefore, the **system-max dvmrp-max-int-group** and the **system-max pim-max-int-group** commands that define their maximum table sizes have been removed.

The software allocates memory globally for each group, and also allocates memory separately for each interface's IGMP membership in a multicast group. An interface becomes a member of a multicast group when the interface receives an IGMP group membership report. For example, if the Layer 3 Switch learns about one multicast group, global memory for one group is used. In addition, if three interfaces on the device receive IGMP group membership reports for the group, interface memory for three IGMP memberships also is used.

Since the same group can use multiple allocations of memory (one for the group itself and one for each interface's membership in the group), you can increase the maximum number of IGMP memberships, up to 8192.

NOTE: The total for IGMP memberships applies to the device, not to individual interfaces. You can have up to 8192 IGMP memberships on all the individual interfaces, not up to 8192 IGMP memberships on each interface.

Increasing the Number of IGMP Memberships

To increase the number of IGMP membership interfaces for PIM, enter commands such as the following:

```
FastIron(config)#system-max pim-max-int-group 4000
FastIron(config)#write memory
```

This command enables the device to have up to 4000 IGMP memberships for PIM.

NOTE: The **system-max pim-max-int-group** command is no longer available since you can configure an unlimited number of PIM interface groups for DVMRP.

Syntax: [no] system-max pim-max-int-group <num>

The <num> parameter specifies the maximum number of IGMP memberships for PIM, and can be from 256 – 8192.

To increase the number of IGMP memberships interfaces you can have for DVMRP, enter commands such as the following:

```
FastIron(config)#system-max dvmrp-max-int-group 3000
FastIron(config)#write memory
```

NOTE: The **system-max dvmrp-max-int-group** command is no longer available since you can configure an unlimited number of DVMRP interface groups.

Syntax: [no] system-max dvmrp-max-int-group <num>

The <num> parameter specifies the maximum number of IGMP memberships for DVMRP, and can be from 256 – 8192.

NOTE: You do not need to reload the software for these changes to take effect.

Defining the Maximum Number of DVMRP Cache Entries

The DVMRP cache system parameter defines the maximum number of repeated DVMRP traffic being sent from the same source address and being received by the same destination address. To define this maximum, enter a command such as the following:

```
FastIron(config)#system-max dvmrp-mcache 500
```

Syntax: system-max dvmrp-mcache <num>

The <num> parameter specifies the maximum number of multicast cache entries for DVMRP. Enter a number from 128 – 2048. The default is 512.

Defining the Maximum Number of PIM Cache Entries

The PIM cache system parameter defines the maximum number of repeated PIM traffic being sent from the same source address and being received by the same destination address. To define this maximum, enter a command such as the following:

```
FastIron(config)#system-max pim-mcache 999
```

Syntax: system-max pim-mcache <num>

The <num> parameter specifies the maximum number of multicast cache entries for PIM. Enter a number from 256 – 4096. The default is 1024.

Changing IGMP V1 and V2 Parameters

IGMP allows Foundry routers to limit the multicast of IGMP packets to only those ports on the router that are identified as IP Multicast members. This section applies to Foundry devices that support IGMP versions 1 and 2.

The router actively sends out host queries to identify IP Multicast groups on the network, inserts the group information in an IGMP packet, and forwards the packet to IP Multicast neighbors.

The following IGMP V1 and V2 parameters apply to PIM and DVMRP:

- IGMP query interval – Specifies how often the Layer 3 Switch queries an interface for group membership. Possible values are 1 – 3600. The default is 60.
- IGMP group membership time – Specifies how many seconds an IP Multicast group can remain on a Layer 3 Switch interface in the absence of a group report. Possible values are 1 – 7200. The default is 60.
- IGMP maximum response time – Specifies how many seconds the Layer 3 Switch will wait for an IGMP response from an interface before concluding that the group member on that interface is down and removing the interface from the group. Possible values are 1 – 10. The default is 5.

To change these parameters, you must first enable IP multicast routing by entering the following CLI command at the global CLI level:

```
FastIron(config)#ip multicast-routing
```

Syntax: [no] ip multicast-routing

NOTE: You must enter the **ip multicast-routing** command before changing the global IP Multicast parameters. Otherwise, the changes do not take effect and the software uses the default values.

Modifying IGMP (V1 and V2) Query Interval Period

The IGMP query interval period defines how often a router will query an interface for group membership. Possible values are 1 – 3,600 seconds and the default value is 60 seconds.

To modify the default value for the IGMP (V1 and V2) query interval, enter the following:

```
FastIron(config)#ip igmp query 120
```

Syntax: ip igmp query-interval <1-3600>

Modifying IGMP (V1 and V2) Membership Time

Group membership time defines how long a group will remain active on an interface in the absence of a group report. Possible values are from 1 – 7200 seconds and the default value is 140 seconds.

To define an IGMP (V1 and V2) membership time of 240 seconds, enter the following:

```
FastIron(config)#ip igmp group-membership-time 240
```

Syntax: ip igmp group-membership-time <1-7200>

Modifying IGMP (V1 and V2) Maximum Response Time

Maximum response time defines how long the Layer 3 Switch will wait for an IGMP (V1 and V2) response from an interface before concluding that the group member on that interface is down and removing the interface from the group. Possible values are 1 – 10. The default is 5.

To change the IGMP (V1 and V2) maximum response time, enter a command such as the following at the global CONFIG level of the CLI:

```
FastIron(config)#ip igmp max-response-time 8
```

Syntax: [no] ip igmp max-response-time <num>

The <num> parameter specifies the number of seconds and can be a value from 1 – 10. The default is 5.

Adding an Interface to a Multicast Group

You can manually add an interface to a multicast group. This is useful in the following cases:

- Hosts attached to the interface are unable to add themselves as members of the group using IGMP.
- There are no members for the group attached to the interface.

When you manually add an interface to a multicast group, the Foundry device forwards multicast packets for the group but does not itself accept packets for the group.

You can manually add a multicast group to individual ports only. If the port is a member of a virtual routing interface, you must add the ports to the group individually.

To manually add a port to a multicast group, enter a command such as the following at the configuration level for the port:

```
FastIron(config-if-1/1)#ip igmp static-group 224.2.2.2
```

This command adds port 1/1 to multicast group 224.2.2.2.

To add a port that is a member of a virtual routing interface to a multicast group, enter a command such as the following at the configuration level for the virtual routing interface:

```
FastIron(config-vif-1)#ip igmp static-group 224.2.2.2 ethernet 5/2
```

This command adds port 5/2 in virtual routing interface 1 to multicast group 224.2.2.2.

Syntax: [no] ip igmp static-group <ip-addr> [ethernet <portnum>]

The <ip-addr> parameter specifies the group number.

The **ethernet** <portnum> parameter specifies the port number. Use this parameter if the port is a member of a virtual routing interface, and you are entering this command at the configuration level for the virtual routing interface.

Manually added groups are included in the group information displayed by the following commands:

- **show ip igmp group**
- **show ip pim group**

PIM Dense

NOTE: This section describes the “dense” mode of PIM, described in RFC 1075. See “PIM Sparse” on page 30-12 for information about PIM Sparse.

PIM was introduced to simplify some of the complexity of the routing protocol at the cost of additional overhead tied with a greater replication of forwarded multicast packets. PIM is similar to DVMRP in that PIM builds source-routed multicast delivery trees and employs reverse path check when forwarding multicast packets.

There are two modes in which PIM operates: Dense and Sparse. The Dense Mode is suitable for densely populated multicast groups, primarily in the LAN environment. The Sparse Mode is suitable for sparsely populated multicast groups with the focus on WAN.

PIM primarily differs from DVMRP by using the IP routing table instead of maintaining its own, thereby being routing protocol independent.

Initiating PIM Multicasts on a Network

Once PIM is enabled on each router, a network user can begin a video conference multicast from the server on R1 as shown in Figure 30.1. When a multicast packet is received on a PIM-capable router interface, the interface checks its IP routing table to determine whether the interface that received the message provides the shortest path back to the source. If the interface does provide the shortest path back to the source, the multicast packet is then forwarded to all neighboring PIM routers. Otherwise, the multicast packet is discarded and a prune message is sent back upstream.

In Figure 30.1, the root node (R1) is forwarding multicast packets for group 229.225.0.1, which it receives from the server, to its downstream nodes, R2, R3, and R4. Router R4 is an intermediate router with R5 and R6 as its downstream routers. Because R5 and R6 have no downstream interfaces, they are leaf nodes. The receivers in this example are those workstations that are resident on routers R2, R3, and R6.

Pruning a Multicast Tree

As multicast packets reach these leaf routers, the routers check their IGMP databases for the group. If the group is not in a router's IGMP database, the router discards the packet and sends a prune message to the upstream router. The router that discarded the packet also maintains the prune state for the source, group (S,G) pair. The branch is then pruned (removed) from the multicast tree. No further multicast packets for that specific (S,G) pair will be received from that upstream router until the prune state expires. You can configure the PIM Prune Timer (the length of time that a prune state is considered valid).

For example, in Figure 30.1 the sender with address 207.95.5.1 is sending multicast packets to the group 229.225.0.1. If a PIM switch receives any groups other than that group, the switch discards the group and sends a prune message to the upstream PIM switch.

In Figure 30.2, switch S5 is a leaf node with no group members in its IGMP database. Therefore, the switch must be pruned from the multicast tree. S5 sends a prune message upstream to its neighbor switch S4 to remove itself from the multicast delivery tree and install a prune state, as seen in Figure 30.2. Switch S5 will not receive any further multicast traffic until the prune age interval expires.

When a node on the multicast delivery tree has all of its downstream branches (downstream interfaces) in the prune state, a prune message is sent upstream. In the case of S4, if both S5 and S6 are in a prune state at the same time, S4 becomes a leaf node with no downstream interfaces and sends a prune message to S1. With S4 in a prune state, the resulting multicast delivery tree would consist only of leaf nodes S2 and S3.

Figure 30.1 Transmission of Multicast Packets from the Source to Host Group Members

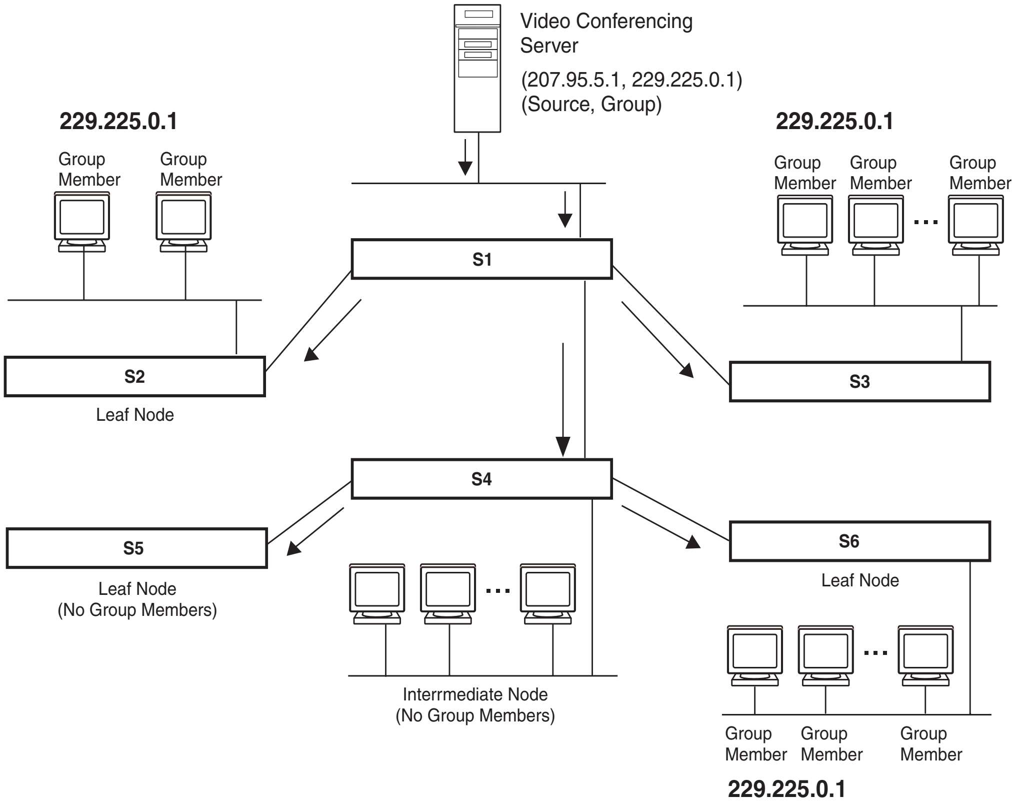
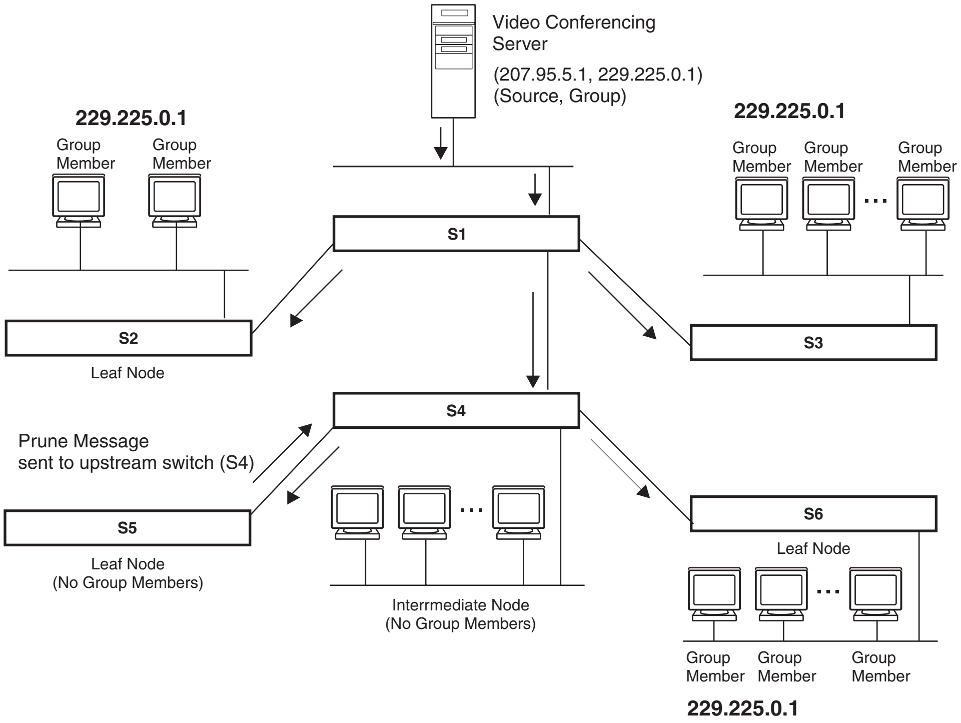


Figure 30.2 Pruning Leaf Nodes from a Multicast Tree



Grafts to a Multicast Tree

A PIM switch restores pruned branches to a multicast tree by sending graft messages towards the upstream switch. Graft messages start at the leaf node and travel up the tree, first sending the message to its neighbor upstream switch.

In the example above, if a new 229.255.0.1 group member joins on switch S6, which was previously pruned, a graft is sent upstream to S4. Since the forwarding state for this entry is in a prune state, S4 sends a graft to S1. Once S4 has joined the tree, S4 and S6 once again receive multicast packets.

Prune and graft messages are continuously used to maintain the multicast delivery tree. No configuration is required on your part.

PIM DM Versions

Foundry devices support PIM DM V1 and V2. The default is V2. You can specify the version on an individual interface basis.

The primary difference between PIM DM V1 and V2 is the methods the protocols use for messaging:

- PIM DM V1 – uses the Internet Group Management Protocol (IGMP) to send messages
- PIM DM V2 – sends messages to the multicast address 224.0.0.13 (ALL-PIM-ROUTERS) with protocol number 103

The CLI commands for configuring and managing PIM DM are the same for V1 and V2. The only difference is the command you use to enable the protocol on an interface.

NOTE: Version 2 is the default PIM DM version. The only difference between version 1 and version 2 is the way the protocol sends messages. The change is not apparent in most configurations. You can use version 2 instead of version 1 with no impact to your network. However, if you want to continue to use PIM DM V1 on an interface, you must change the version, then save the configuration.

NOTE: The note above doesn't mean you can run different PIM versions on devices that are connected to each other. The devices must run the same version of PIM. If you want to connect a Layer 3 Switch running PIM to a device that is running PIM V1, you must change the version on the Layer 3 Switch to V1 (or change the version on the device to V2, if supported).

Configuring PIM DM

NOTE: This section describes how to configure the "dense" mode of PIM, described in RFC 1075. See "Configuring PIM Sparse" on page 30-14 for information about configuring PIM Sparse.

Enabling PIM on the Router and an Interface

By default, PIM is disabled. To enable PIM:

- Enable the feature globally.
- Configure the IP interfaces that will use PIM.
- Enable PIM locally on the ports that have the IP interfaces you configured for PIM.
- Reload the software to place PIM into effect.

Suppose you want to initiate the use of desktop video for fellow users on a sprawling campus network. All destination workstations have the appropriate hardware and software but the Foundry routers that connect the various buildings need to be configured to support PIM multicasts from the designated video conference server as shown in Figure 30.1 on page 30-6.

PIM is enabled on each of the Foundry routers shown in Figure 30.1, on which multicasts are expected. You can enable PIM on each router independently or remotely from one of the routers with a Telnet connection. Follow the same steps for each router. A reset of the router is required when PIM is first enabled. Thereafter, all changes are dynamic.

Globally Enabling and Disabling PIM

To globally enable PIM, enter the following command:

```
FastIron(config)#router pim
```

Syntax: [no] router pim

The behavior of the **[no] router pim** command is as follows:

- Entering **router pim** command to enable PIM does not require a software reload.
- Entering a **no router pim** command removes all configuration for PIM multicast on a Layer 3 Switch (**router pim** level) only.

Globally Enabling and Disabling PIM without Deleting Multicast Configuration

As stated above entering a **no router pim** command deletes the PIM configuration. If you want to disable PIM without deleting any PIM configuration, enter the following command:

```
FastIron(config)#router pim
FastIron(config-pim-router)#disable-pim
```

Syntax: [no] disable-pim

Use the [no] version of the command to re-enable PIM.

Enabling a PIM version

USING THE CLI

To enable PIM on an interface, globally enable PIM, then enable PIM on interface 3, enter the following commands:

```
FastIron(config)#router pim
FastIron(config)#int e 3
FastIron(config-if-e1000-3)#ip address 207.95.5.1/24
FastIron(config-if-e1000-3)#ip pim
FastIron(config-if-e1000-3)#write memory
FastIron(config-if-e1000-3)#end
FastIron#reload
```

Syntax: [no] ip pim [version 1 | 2]

The **version 1 | 2** parameter specifies the PIM DM version. The default version is 2.

If you have enabled PIM version 1 but need to enable version 2 instead, enter either of the following commands at the configuration level for the interface:

```
FastIron(config-if-1/1)#ip pim version 2
```

```
FastIron(config-if-1/1)#no ip pim version 1
```

To disable PIM DM on the interface, enter the following command:

```
FastIron(config-if-1/1)#no ip pim
```

Modifying PIM Global Parameters

PIM global parameters come with preset values. The defaults work well in most networks, but you can modify the following parameters if you need to:

- Neighbor timeout
- Hello timer
- Prune timer
- Prune wait timer
- Graft retransmit timer
- Inactivity timer

Modifying Neighbor Timeout

Neighbor timeout is the interval after which a PIM router will consider a neighbor to be absent. Absence of PIM hello messages from a neighboring router indicates that a neighbor is not present.

The default value is 180 seconds.

To apply a PIM neighbor timeout value of 360 seconds to all ports on the router operating with PIM, enter the following:

```
FastIron(config)#router pim
FastIron(config-pim-router)#nbr-timeout 360
```

Syntax: nbr-timeout <60-8000>

The default is 180 seconds.

Modifying Hello Timer

This parameter defines the interval at which periodic hellos are sent out PIM interfaces. Routers use hello messages to inform neighboring routers of their presence. The default rate is 60 seconds.

To apply a PIM hello timer of 120 seconds to all ports on the router operating with PIM, enter the following:

```
FastIron(config)#router pim
FastIron(config-pim-router)#hello-timer 120
```

Syntax: hello-timer <10-3600>

The default is 60 seconds.

Modifying Prune Timer

This parameter defines how long a Foundry PIM router will maintain a prune state for a forwarding entry.

The first received multicast interface is forwarded to all other PIM interfaces on the router. If there is no presence of groups on that interface, the leaf node sends a prune message upstream and stores a prune state. This prune state travels up the tree and installs a prune state.

A prune state is maintained until the prune timer expires or a graft message is received for the forwarding entry. The default value is 180 seconds.

To set the PIM prune timer to 90, enter the following:

```
FastIron(config)#router pim
FastIron(config-pim-router)##prune-timer 90
```

Syntax: prune-timer <10-3600>

The default is 180 seconds.

Modifying the Prune Wait Timer

The CLI command **prune-wait** allows you to configure the amount of time a PIM router will wait before stopping traffic to neighbor routers that do not want the traffic. The value can be from zero to three seconds. The default is three seconds. A smaller prune wait value reduces flooding of unwanted traffic.

A prune wait value of zero causes the PIM router to stop traffic immediately upon receiving a prune message. If there are two or more neighbors on the physical port, then the **prune-wait** command should not be used because one neighbor may send a prune message while the other sends a join message at the during time or in less than three seconds.

To set the prune wait time to zero, enter the following commands:

```
FastIron(config)#router pim
FastIron(config-pim-router)#prune-wait 0
```

Syntax: prune-wait <time>

where <time> can be 0 - 3 seconds. A value of 0 causes the PIM router to stop traffic immediately upon receiving a prune message. The default is 3 seconds.

Viewing the Prune Wait Time

To view the prune wait time, enter the **show ip pim dense** command at any level of the CLI.

```
FastIron#show ip pim dense

Global PIM Dense Mode Settings
Hello interval: 60, Neighbor timeout: 180
Graft Retransmit interval: 180, Inactivity interval: 180
Route Expire interval: 200, Route Discard interval: 340
Prune age: 180, Prune wait: 3
```

Modifying Graft Retransmit Timer

The Graft Retransmit Timer defines the interval between the transmission of graft messages.

A graft message is sent by a router to cancel a prune state. When a router receives a graft message, the router responds with a Graft Ack (acknowledge) message. If this Graft Ack message is lost, the router that sent the graft message will resend it.

To change the graft retransmit timer from the default of 180 to 90 seconds, enter the following:

```
FastIron(config)#router pim
FastIron(config-pim-router)#graft-retransmit-timer 90
```

Syntax: graft-retransmit-timer <10-3600>

The default is 180 seconds.

Modifying Inactivity Timer

The router deletes a forwarding entry if the entry is not used to send multicast packets. The PIM inactivity timer defines how long a forwarding entry can remain unused before the router deletes it.

To apply a PIM inactivity timer of 90 seconds to all PIM interfaces, enter the following:

```
FastIron(config)#router pim
FastIron(config-pim-router)#inactivity-timer 90
```

Syntax: inactivity-timer <10-3600>

The default is 180 seconds.

Selection of Shortest Path Back to Source

By default, when a multicast packet is received on a PIM-capable router interface in a multi-path topology, the interface checks its IP routing table to determine the shortest path back to the source. If the alternate paths have the same cost, the first alternate path in the table is picked as the path back to the source. For example, in the table below, the first four routes have the same cost back to the source. However, 137.80.127.3 will be chosen as the path to the source since it is the first one on the list. The router rejects traffic from any port other than Port V11 on which 137.80.127.3 resides.

```
    Total number of IP routes: 19
    B:BGP D:Connected  R:RIP  S:Static  O:OSPF *:Candidate default
            Destination     NetMask         Gateway         Port       Cost
Type
       ..
    9       172.17.41.4     255.255.255.252*137.80.127.3    v11        2
O
            172.17.41.4     255.255.255.252 137.80.126.3    v10        2
O
            172.17.41.4     255.255.255.252 137.80.129.1    v13        2
O
            172.17.41.4     255.255.255.252 137.80.128.3    v12        2
O
    10      172.17.41.8     255.255.255.252 0.0.0.0         1/2        1
D
```

When the Highest IP RPF feature is enabled, the selection of the shortest path back to the source is based on which Reverse Path Forwarding (RPF) neighbor in the IP routing table has the highest IP address, if the cost of the routes are the same. For example, in the table above, Gateway 137.80.129.1 will be chosen as the shortest path to the source because it is the RPF neighbor with the highest IP address.

When choosing the RPF, the router first checks the Multicast Routing Table. If the table is not available, it chooses an RPF from the IP Routing Table. Multicast route is configured using the **ip mroute** command.

To enable the Highest IP RPF feature, enter commands such as the following:

```
FastIron(config)#router pim
FastIron(config-pim-router)#highest-ip-rpf
```

The command immediately enables the Highest IP RPF feature; there is no need to reboot the device.

Syntax: [no] highest-ip-rpf

Entering the **no** version of the command disables the feature; the shortest path back to the source will be based on the first entry in the IP routing table. If some PIM traffic paths were selected based on the highest IP RPF, these paths are changed immediately to use the first RPF in the routing table.

Failover Time in a Multi-Path Topology

When a port in a multi-path topology fails, and the failed port is the input port of the downstream router, a new path is re-established within a few seconds, depending on the routing protocol being used.

No configuration is required for this feature.

Modifying the TTL

The TTL defines the minimum value required in a packet for it to be forwarded out of the interface.

For example, if the TTL for an interface is set at 10, it means that only those packets with a TTL value of 10 or more will be forwarded. Likewise, if an interface is configured with a TTL Threshold value of 1, all packets received on that interface will be forwarded. Possible TTL values are 1 to 31. The default TTL value is 1.

Configuration Notes

- If the TTL for an interface is greater than 1, PIM packets received on the interface are always forwarded in software because each packet's TTL must be examined. Therefore, Foundry does not recommend modifying the TTL under normal operating conditions.
- Multicast packets with a TTL value of 1 are switched within the same VLAN. These packets cannot be routed between different VLANs.

Configuration Syntax

To configure a TTL of 24, enter the following:

```
FastIron(config-if-3/24)#ip pim ttl 24
```

Syntax: ip pim ttl <1-31>

Dropping PIM Traffic in Hardware

Unwanted PIM Dense or PIM Sparse multicast traffic can be dropped in hardware on Layer 3 Switches. This feature does not apply to DVMRP traffic. Refer to “Passive Multicast Route Insertion” on page 30-32.

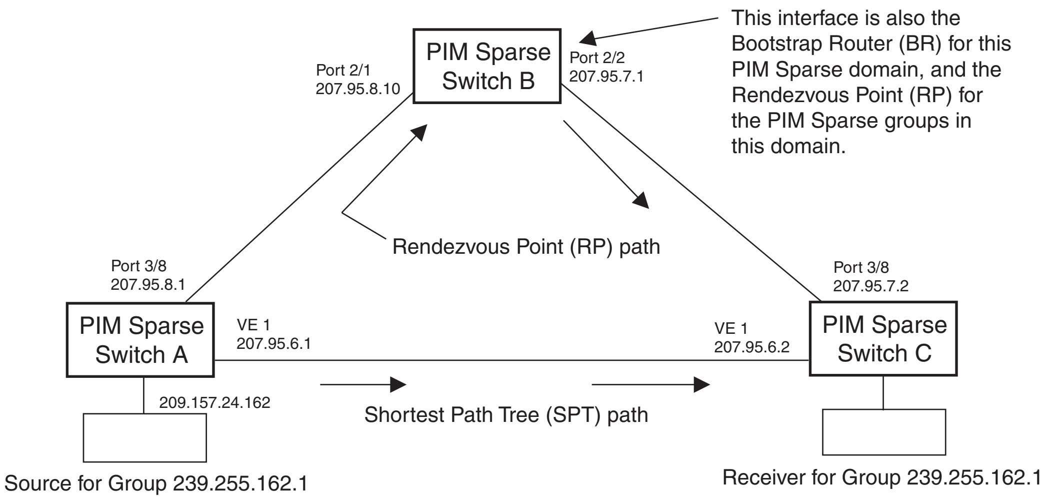
PIM Sparse

Foundry devices support Protocol Independent Multicast (PIM) Sparse version 2. PIM Sparse provides multicasting that is especially suitable for widely distributed multicast environments. The Foundry implementation is based on RFC 2362.

In a PIM Sparse network, a PIM Sparse router that is connected to a host that wants to receive information for a multicast group must explicitly send a join request on behalf of the receiver (host).

PIM Sparse routers are organized into domains. A PIM Sparse domain is a contiguous set of routers that all implement PIM and are configured to operate within a common boundary. Figure 30.3 shows a simple example of a PIM Sparse domain. This example shows three Layer 3 Switches configured as PIM Sparse routers. The configuration is described in detail following the figure.

Figure 30.3 Example of a PIM Sparse Domain


PIM Sparse Switch B
Port 2/1
207.95.8.10
Port 2/2
207.95.7.1
This interface is also the Bootstrap Router (BR) for this PIM Sparse domain, and the Rendezvous Point (RP) for the PIM Sparse groups in this domain.
Rendezvous Point (RP) path
Port 3/8
207.95.8.1
Port 3/8
207.95.7.2
PIM Sparse Switch A
VE 1
207.95.6.1
VE 1
207.95.6.2
PIM Sparse Switch C
209.157.24.162
Shortest Path Tree (SPT) path
Source for Group 239.255.162.1
Receiver for Group 239.255.162.1

PIM Sparse Switch Types

Switches that are configured with PIM Sparse interfaces also can be configured to fill one or more of the following roles:

- PMBR – A PIM switch that has some interfaces within the PIM domain and other interface outside the PIM domain. PBMRs connect the PIM domain to the Internet.

NOTE: You cannot configure a Foundry routing interface as a PMBR interface for PIM Sparse in the current software release.

- BSR – The Bootstrap Router (BSR) distributes RP information to the other PIM Sparse switches within the domain. Each PIM Sparse domain has one active BSR. For redundancy, you can configure ports on multiple switches as candidate BSRs. The PIM Sparse protocol uses an election process to select one of the candidate BSRs as the BSR for the domain. The BSR with the highest BSR priority (a user-configurable parameter) is elected. If the priorities result in a tie, then the candidate BSR interface with the highest IP address is elected**.** In the example in Figure 30.3, PIM Sparse switch B is the BSR. Port 2/2 is configured as a candidate BSR.
- RP – The RP is the meeting point for PIM Sparse sources and receivers. A PIM Sparse domain can have multiple RPs, but each PIM Sparse multicast group address can have only one active RP. PIM Sparse switches learn the addresses of RPs and the groups for which they are responsible from messages that the BSR sends to each of the PIM Sparse switches. In the example in Figure 30.3, PIM Sparse Switch B is the RP. Port 2/2 is configured as a candidate Rendezvous Point (RP).

  To enhance overall network performance, Foundry Layer 3 Switches use the RP to forward only the first packet from a group source to the group's receivers. After the first packet, the Layer 3 Switch calculates the shortest path between the receiver and source (the Shortest Path Tree, or SPT) and uses the SPT for subsequent packets from the source to the receiver. The Layer 3 Switch calculates a separate SPT for each source-receiver pair.

NOTE: Foundry Networks recommends that you configure the same ports as candidate BSRs and RPs.

RP Paths and SPT Paths

Figure 30.3 shows two paths for packets from the source for group 239.255.162.1 and a receiver for the group. The source is attached to PIM Sparse Switch A and the recipient is attached to PIM Sparse Switch C. PIM Sparse Switch B in is the RP for this multicast group. As a result, the default path for packets from the source to the receiver is through the RP. However, the path through the RP sometimes is not the shortest path. In this case, the shortest path between the source and the receiver is over the direct link between Switch A and Switch C, which bypasses the RP (Switch B).

To optimize PIM traffic, the protocol contains a mechanism for calculating the Shortest Path Tree (SPT) between a given source and receiver. PIM Sparse switches can use the SPT as an alternative to using the RP for forwarding traffic from a source to a receiver. By default, Foundry Layer 3 Switches forward the first packet they receive from a given source to a given receiver using the RP path, but forward subsequent packets from that source to that receiver through the SPT. In Figure 30.3, Switch A forwards the first packet from group 239.255.162.1's source to the destination by sending the packet to Switch B, which is the RP. Switch B then sends the packet to Switch C. For the second and all future packets that Switch A receives from the source for the receiver, Switch A forwards them directly to Switch C using the SPT path.

Configuring PIM Sparse

To configure a Foundry Layer 3 Switch for PIM Sparse, perform the following tasks:

- Configure the following global parameter:
  - Enable the PIM Sparse mode of multicast routing.
- Configure the following interface parameters:
  - Configure an IP address on the interface
  - Enable PIM Sparse.
  - Identify the interface as a PIM Sparse border, if applicable.

NOTE: You cannot configure a Foundry routing interface as a PMBR interface for PIM Sparse in the current software release.

- Configure the following PIM Sparse global parameters:
  - Identify the Layer 3 Switch as a candidate PIM Sparse Bootstrap Router (BSR), if applicable.
  - Identify the Layer 3 Switch as a candidate PIM Sparse Rendezvous Point (RP), if applicable.
  - Specify the IP address of the RP (if you want to statically select the RP).

NOTE: Foundry Networks recommends that you configure the same Layer 3 Switch as both the BSR and the RP.

Limitations in this Release

The implementation of PIM Sparse in the current software release has the following limitations:

- PIM Border Routers (PMBRs) are not supported. Thus, you cannot configure a Foundry routing interface as a PMBR interface for PIM Sparse.
- PIM Sparse and regular PIM (dense mode) cannot be used on the same interface.
- You cannot configure or display PIM Sparse information using the Web management interface. (You can display some general PIM information, but not specific PIM Sparse information.)

Configuring Global PIM Sparse Parameters

To configure the PIM Sparse global parameters, use either of the following methods.

To configure basic global PIM Sparse parameters, enter commands such as the following on each Layer 3 Switch within the PIM Sparse domain:

```
FastIron(config)#router pim
```

Syntax: [no] router pim

NOTE: You do not need to globally enable IP multicast routing when configuring PIM Sparse.

The command in this example enables IP multicast routing, and enables the PIM Sparse mode of IP multicast routing. The command does not configure the Layer 3 Switch as a candidate PIM Sparse Bootstrap Router (BSR) and candidate Rendezvous Point (RP). You can configure a Foundry Layer 3 Switch as a PIM Sparse switch without configuring the it as a candidate BSR and RP. However, if you do configure the Layer 3 Switch as one of these, Foundry Networks recommends that you configure it as both. See “Configuring BSRs” on page 30-15.

The behavior of the **[no] router pim** command is as follows:

- Entering **no router pim** command to disable PIM or DVMRP does not require a software reload.
- Entering a **no router pim** command removes all configuration for PIM multicast on a Layer 3 Switch (**router pim** level) only.

Globally Enabling and Disabling PIM without Deleting the Multicast Configuration

As stated above entering a **no router pim** command deletes the PIM configuration. If you want to disable PIM without deleting any PIM configuration, enter the following command:

```
FastIron(config)#router pim
FastIron(config-pim-router)#disable-pim
```

Syntax: [no] disable-pim

Use the [no] version of the command to re-enable PIM.

Configuring PIM Interface Parameters

After you enable IP multicast routing and PIM Sparse at the global level, you must enable it on the individual interfaces connected to the PIM Sparse network. To do so, use the following CLI method.

To enable PIM Sparse mode on an interface, enter commands such as the following:

```
FastIron(config)#interface ethernet 2/2
FastIron(config-if-2/2)#ip address 207.95.7.1 255.255.255.0
FastIron(config-if-2/2)#ip pim-sparse
```

Syntax: [no] ip pim-sparse

The commands in this example add an IP interface to port 2/2, then enable PIM Sparse on the interface.

If the interface is on the border of the PIM Sparse domain, you also must enter the following command:

```
FastIron(config-if-2/2)#ip pim border
```

Syntax: [no] ip pim border

NOTE: You cannot configure a Foundry routing interface as a PMBR interface for PIM Sparse in the current software release.

Configuring BSRs

In addition to the global and interface parameters in the sections above, you need to identify an interface on at least one Layer 3 Switch as a candidate PIM Sparse Bootstrap router (BSR) and candidate PIM Sparse Rendezvous Point (RP).

NOTE: It is possible to configure the Layer 3 Switch as only a candidate BSR or RP, but Foundry Networks recommends that you configure the same interface on the same Layer 3 Switch as both a BSR and an RP.

This section presents how to configure BSRs. Refer to “Configuring RPs” on page 30-16 for instructions on how to configure RPs.

To configure the Layer 3 Switch as a candidate BSR and RP, enter commands such as the following:

```
FastIron(config)#router pim
FastIron(config-pim-router)#bsr-candidate ethernet 2/2 30 255
BSR address: 207.95.7.1, hash mask length: 30, priority: 255
```

This command configures the PIM Sparse interface on port 2/2 as a BSR candidate, with a hash mask length of 30 and a priority of 255. The information shown in italics above is displayed by the CLI after you enter the candidate BSR configuration command.

Syntax: [no] bsr-candidate ethernet [<slotnum>/]<portnum> | loopback <num> | ve <num> <hash-mask-length> [<priority>]

The <slotnum> parameter is required on chassis devices.

The <portnum> | **loopback** <num> | **ve** <num> parameter specifies the interface. The Layer 3 Switch will advertise the specified interface's IP address as a candidate BSR.

- Enter **ethernet** [<slotnum>/] <portnum> for a physical interface (port).
- Enter **ve** <num> for a virtual interface.
- Enter **loopback** <num> for a loopback interface.

The <hash-mask-length> parameter specifies the number of bits in a group address that are significant when calculating the group-to-RP mapping. You can specify a value from 1 – 32.

NOTE: Foundry Networks recommends you specify 30 for IP version 4 (IPv4) networks.

The <priority> specifies the BSR priority. You can specify a value from 0 – 255. When the election process for BSR takes place, the candidate BSR with the highest priority becomes the BSR. The default is 0.

Configuring RPs

Enter a command such as the following to configure the Layer 3 Switch as a candidate RP:

```
FastIron(config-pim-router)#rp-candidate ethernet 2/2
```

Syntax: [no] rp-candidate ethernet [<slotnum>/]<portnum> | loopback <num> | ve <num>

The <slotnum> parameter is required on chassis devices.

The <portnum> | **loopback** <num> | **ve** <num> parameter specifies the interface. The Layer 3 Switch will advertise the specified interface's IP address as a candidate RP.

- Enter **ethernet** [<slotnum>/]<portnum> for a physical interface (port).
- Enter **ve** <num> for a virtual interface.
- Enter **loopback** <num> for a loopback interface.

By default, this command configures the Layer 3 Switch as a candidate RP for all group numbers beginning with 224. As a result, the Layer 3 Switch is a candidate RP for all valid PIM Sparse group numbers. You can change this by adding or deleting specific address ranges. The following example narrows the group number range for which the Layer 3 Switch is a candidate RP by explicitly adding a range.

```
FastIron(config-pim-router)#rp-candidate add 224.126.0.0 16
```

Syntax: [no] rp-candidate add <group-addr> <mask-bits>

The <group-addr> <mask-bits> specifies the group address and the number of significant bits in the subnet mask. In this example, the Layer 3 Switch is a candidate RP for all groups that begin with 224.126. When you add a range, you override the default. The Layer 3 Switch then becomes a candidate RP only for the group address range(s) you add.

You also can change the group numbers for which the Layer 3 Switch is a candidate RP by deleting address ranges. For example, to delete all addresses from 224.126.22.0 – 224.126.22.255, enter the following command:

```
FastIron(config-pim-router)#rp-candidate delete 224.126.22.0 24
```

Syntax: [no] rp-candidate delete <group-addr> <mask-bits>

The usage of the <group-addr> <mask-bits> parameter is the same as for the **rp-candidate add** command.

If you enter both commands shown in the example above, the net effect is that the Layer 3 Switch becomes a candidate RP for groups 224.126.0.0 – 224.126.21.255 and groups 224.126.23.0 – 224.126.255.255.

Updating PIM-Sparse Forwarding Entries with New RP Configuration

If you make changes to your static RP configuration, the entries in the PIM-Sparse multicast forwarding table continue to use the old RP configuration until they are aged out.

The **clear pim rp-map** command allows you to update the entries in the static multicast forwarding table immediately after making RP configuration changes. This command is meant to be used with **rp-address** command.

To update the entries in a PIM sparse static multicast forwarding table with new RP configuration, enter the following command at the privileged EXEC level of the CLI:

```
FastIron#clear pim rp-map
```

Syntax: clear pim rp-map

Statically Specifying the RP

Foundry Networks recommends that you use the PIM Sparse protocol’s RP election process so that a backup RP can automatically take over if the active RP router becomes unavailable. However, if you do not want the RP to be selected by the RP election process but instead you want to explicitly identify the RP by its IP address, you can do using the following CLI method.

If you explicitly specify the RP, the Layer 3 Switch uses the specified RP for all group-to-RP mappings and overrides the set of candidate RPs supplied by the BSR.

NOTE: Specify the same IP address as the RP on all PIM Sparse routers within the PIM Sparse domain. Make sure the router is on the backbone or is otherwise well connected to the rest of the network.

To specify the IP address of the RP, enter commands such as the following:

```
FastIron(config)#router pim
FastIron(config-pim-router)#rp-address 207.95.7.1
```

Syntax: [no] rp-address <ip-addr>

The <ip-addr> parameter specifies the IP address of the RP.

The command in the example above identifies the router interface at IP address 207.95.7.1 as the RP for the PIM Sparse domain. The Layer 3 Switch will use the specified RP and ignore group-to-RP mappings received from the BSR.

Changing the Shortest Path Tree (SPT) Threshold

In a typical PIM Sparse domain, there may be two or more paths from a DR (designated router) for a multicast source to a PIM group receiver.

- **Path through the RP** – This is the path the Layer 3 Switch uses the first time it receives traffic for a PIM group. However, the path through the RP may not be the shortest path from the Layer 3 Switch to the receiver.
- **Shortest Path** – Each PIM Sparse router that is a DR for a multicast source calculates a shortest path tree (SPT) to all the PIM Sparse group receivers within the domain, with the Layer 3 Switch itself as the root of the tree. The first time a Foundry Layer 3 Switch configured as a PIM router receives a packet for a PIM receiver, the Layer 3 Switch sends the packet to the RP for the group. The Layer 3 Switch also calculates the SPT from itself to the receiver. The next time the Layer 3 Switch receives a PIM Sparse packet for the receiver, the Layer 3 Switch sends the packet toward the receiver using the shortest route, which may not pass through the RP.

By default, the device switches from the RP to the SPT after receiving the first packet for a given PIM Sparse group. The Layer 3 Switch maintains a separate counter for each PIM Sparse source-group pair.

After the Layer 3 Switch receives a packet for a given source-group pair, the Layer 3 Switch starts a PIM data timer for that source-group pair. If the Layer 3 Switch does not receive another packet for the source-group pair before the timer expires, it reverts to using the RP for the next packet received for the source-group pair. In accordance with the PIM Sparse RFC's recommendation, the timer is 210 seconds and is not configurable. The counter is reset to zero each time the Layer 3 Switch receives a packet for the source-group pair.

You can change the number of packets that the Layer 3 Switch sends using the RP before switching to using the SPT. To do so, use the following CLI method.

```
FastIron(config)#router pim
FastIron(config-pim-router)#spt-threshold 1000
```

Syntax: [no] spt-threshold infinity | <num>

The **infinity** | <num> parameter specifies the number of packets. If you specify **infinity**, the Layer 3 Switch sends packets using the RP indefinitely and does not switch over to the SPT. If you enter a specific number of packets, the Layer 3 Switch does not switch over to using the SPT until it has sent the number of packets you specify using the RP.

Changing the PIM Join and Prune Message Interval

By default, the Layer 3 Switch sends PIM Sparse Join/Prune messages every 60 seconds. These messages inform other PIM Sparse routers about clients who want to become receivers (Join) or stop being receivers (Prune) for PIM Sparse groups.

You can change the Join/Prune message interval using the following CLI method.

NOTE: Use the same Join/Prune message interval on all the PIM Sparse routers in the PIM Sparse domain. If the routers do not all use the same timer interval, the performance of PIM Sparse can be adversely affected.

To change the Join/Prune interval, enter commands such as the following:

```
FastIron(config)#router pim
FastIron(config-pim-router)#message-interval 30
```

Syntax: [no] message-interval <num>

The <num> parameter specifies the number of seconds and can from 1 – 65535. The default is 60.

Dropping PIM Traffic in Hardware

Unwanted PIM Dense or PIM Sparse multicast traffic can be dropped in hardware on Layer 3 Switches. This feature does not apply to DVMRP traffic. Refer to “Passive Multicast Route Insertion” on page 30-32.

Displaying PIM Sparse Configuration Information and Statistics

You can display the following PIM Sparse information:

- Basic PIM Sparse configuration information
- Group information
- BSR information
- Candidate RP information
- RP-to-group mappings
- RP information for a PIM Sparse group
- RP set list
- PIM Neighbor information
- The PIM flow cache

- The PIM multicast cache
- PIM traffic statistics

Displaying Basic PIM Sparse Configuration Information

To display basic configuration information for PIM Sparse, enter the following command at any CLI level:

```
FastIron#show ip pim sparse

Global PIM Sparse Mode Settings
  Hello interval: 60, Neighbor timeout: 180
  Bootstrap Msg interval: 130, Candidate-RP Advertisement interval: 60
  Join/Prune interval: 60, SPT Threshold: 1

Interface Ethernet e3/8
TTL Threshold: 1, Enabled
Local Address: 207.95.8.1

Interface Ve 1
TTL Threshold: 1, Enabled
Local Address: 207.95.6.1
```

Syntax: show ip pim sparse

This example shows the PIM Sparse configuration information on PIM Sparse router A in Figure 30.3.

This display shows the following information.

Table 30.1: Output of show ip pim sparse

This Field...	Displays...
Global PIM Sparse mode settings	
Hello interval	How frequently the Layer 3 Switch sends PIM Sparse hello messages to its PIM Sparse neighbors. This field show the number of seconds between hello messages. PIM Sparse routers use hello messages to discover one another.
Neighbor timeout	How many seconds the Layer 3 Switch will wait for a hello message from a neighbor before determining that the neighbor is no longer present and removing cached PIM Sparse forwarding entries for the neighbor.
Bootstrap Msg interval	How frequently the BSR configured on the Layer 3 Switch sends the RP set to the RPs within the PIM Sparse domain. The RP set is a list of candidate RPs and their group prefixes. A candidate RP's group prefix indicates the range of PIM Sparse group numbers for which it can be an RP. **Note**: This field contains a value only if an interface on the Layer 3 Switch is elected to be the BSR. Otherwise, the field is blank.
Candidate-RP Advertisement interval	How frequently the candidate PR configured on the Layer 3 Switch sends candidate RP advertisement messages to the BSR. **Note**: This field contains a value only if an interface on the Layer 3 Switch is configured as a candidate RP. Otherwise, the field is blank.

Table 30.1: Output of show ip pim sparse

This Field...	Displays...
Join/Prune interval	How frequently the Layer 3 Switch sends PIM Sparse Join/Prune messages for the multicast groups it is forwarding. This field show the number of seconds between Join/Prune messages. The Layer 3 Switch sends Join/Prune messages on behalf of multicast receivers who want to join or leave a PIM Sparse group. When forwarding packets from PIM Sparse sources, the Layer 3 Switch sends the packets only on the interfaces on which it has received join requests in Join/Prune messages for the source's group. You can change the Join/Prune interval if needed. See "Changing the PIM Join and Prune Message Interval" on page 30-18.
SPT Threshold	The number of packets the Layer 3 Switch sends using the path through the RP before switching to using the SPT path.
PIM Sparse interface information **Note**: You also can display IP multicast interface information using the **show ip pim interface** command. However, this command lists all IP multicast interfaces, including regular PIM (dense mode) and DVMRP interfaces. The **show ip pim sparse** command lists only the PIM Sparse interfaces.	
Interface	The type of interface and the interface number. The interface type can be one of the following: • Ethernet • VE The number is either a port number (and slot number if applicable) or the virtual interface (VE) number.
TTL Threshold	Following the TTL threshold value, the interface state is listed. The interface state can be one of the following: • Disabled • Enabled
Local Address	Indicates the IP address configured on the port or virtual interface.

Displaying a List of Multicast Groups

To display a list of the IP multicast groups the Layer 3 Switch is forwarding, enter the following command at any CLI level:

```
FastIron#show ip pim group

Total number of Groups: 2
Index 1         Group 239.255.162.1       Ports e3/11
```

Syntax: show ip pim group

This display shows the following information.

Table 30.2: Output of show ip pim group

This Field...	Displays...
Total number of Groups	Lists the total number of IP multicast groups the Layer 3 Switch is forwarding. **Note**: This list can include groups that are not PIM Sparse groups. If interfaces on the Layer 3 Switch are configured for regular PIM (dense mode) or DVMRP, these groups are listed too.
Index	The index number of the table entry in the display.
Group	The multicast group address
Ports	The Layer 3 Switch ports connected to the receivers of the groups.

Displaying BSR Information

To display BSR information, enter the following command at any CLI level:

```
FastIron#show ip pim bsr

PIMv2 Bootstrap information

This system is the elected Bootstrap Router (BSR)
  BSR address: 207.95.7.1
  Uptime: 00:33:52, BSR priority: 5, Hash mask length: 32
  Next bootstrap message in 00:00:20

Next Candidate-RP-advertisement in 00:00:10
  RP: 207.95.7.1
    group prefixes:
    224.0.0.0 / 4
Candidate-RP-advertisement period: 60
```

This example show information displayed on a Layer 3 Switch that has been elected as the BSR. The following example shows information displayed on a Layer 3 Switch that is not the BSR. Notice that some fields shown in the example above do not appear in the example below.

```
FastIron#show ip pim bsr

PIMv2 Bootstrap information
 local BSR address = 207.95.7.1
 local BSR priority = 5
```

Syntax: show ip pim bsr

This display shows the following information.

Table 30.3: Output of show ip pim bsr

This Field...	Displays...
BSR address or local BSR address	The IP address of the interface configured as the PIM Sparse Bootstrap Router (BSR). **Note**: If the word “local” does not appear in the field, this Layer 3 Switch is the BSR. If the word “local” does appear, this Layer 3 Switch is not the BSR.
Uptime	The amount of time the BSR has been running. **Note**: This field appears only if this Layer 3 Switch is the BSR.
BSR priority or local BSR priority	The priority assigned to the interface for use during the BSR election process. During BSR election, the priorities of the candidate BSRs are compared and the interface with the highest BSR priority becomes the BSR. **Note**: If the word “local” does not appear in the field, this Layer 3 Switch is the BSR. If the word “local” does appear, this Layer 3 Switch is not the BSR.
Hash mask length	The number of significant bits in the IP multicast group comparison mask. This mask determines the IP multicast group numbers for which the Layer 3 Switch can be a BSR. The default is 32 bits, which allows the Layer 3 Switch to be a BSR for any valid IP multicast group number. **Note**: This field appears only if this Layer 3 Switch is the BSR.
Next bootstrap message in	Indicates how many seconds will pass before the BSR sends its next Bootstrap message. **Note**: This field appears only if this Layer 3 Switch is the BSR.
Next Candidate-PR-advertisement message in	Indicates how many seconds will pass before the BSR sends its next candidate PR advertisement message. **Note**: This field appears only if this Layer 3 Switch is the BSR.
RP	Indicates the IP address of the Rendezvous Point (RP). **Note**: This field appears only if this Layer 3 Switch is the BSR.
group prefixes	Indicates the multicast groups for which the RP listed by the previous field is a candidate RP. **Note**: This field appears only if this Layer 3 Switch is the BSR.
Candidate-RP-advertisement period	Indicates how frequently the BSR sends candidate RP advertisement messages. **Note**: This field appears only if this Layer 3 Switch is the BSR.

Displaying Pim Resources

To display the hardware resource information such as hardware allocation, availability, and limit for software data structure, enter the following command.

```
FastIron#show ip pim resource
                              alloc in-use  avail allo-fail up-limit   get-mem
        NBR list                 64      0     64         0      512         0
        timer                   256      0    256         0     4096         0
        pimsm J/P elem            0      0      0         0    48960         0
        pimsm group2rp            0      0      0         0     4096         0
        pimsm L2 reg xmt         64      0     64         0 no-limit         0
        mcache                  256      0    256         0     1024         0
        mcache hash link        997      0    997         0 no-limit         0
        mcache 2nd hash           9      0      9         0      997         0
        graft if no mcache      197      0    197         0 no-limit         0
        pim/dvm global group    256      0    256         0 no-limit         0
        pim/dvmrp prune         128      0    128         0    40960         0
        Output intf-vlan       2000      0   2000         0 no-limit         0
        group hash link          97      0     97         0 no-limit         0
        2D vlan for nbr, glb   2000      0   2000         0 no-limit         0
        Output intf.           1024      0   1024         0 no-limit         0
        2D for glb grp         1024      0   1024         0 no-limit         0
        pim/dvm config. intf    128      2    126         0 no-limit         2
        Prune rate limit        256      0    256         0 no-limit         0
        Distributed add cpu     128      0    128         0 no-limit         0
        L2 VIDX                 256      0    256         0     4096         0
        L2 VIDX hash            997      0    997         0 no-limit         0
        igmp group              256      0    256         0     4096         0
        igmp phy port          1024      0   1024         0 no-limit         0
        igmp exist phy port    1024      4   1020         0 no-limit         4
        igmp G/GS query         128      0    128         0 no-limit         0
        igmp v3 source         2000      0   2000         0   500000         0
        igmp v3 tracking          0      0      0         0 no-limit         0
        igmp glb sorted list   2000      0   2000         0   500000         0
        total pool memory 286918 bytes

        #of PIM ports: physical 2, VEs 0 (max: 512), loopback 0, tunnels 0
        Total Mlls in pool: 943  Allocated MLL: 0  Available MLL: 943
        SW processed pkts 0
```

Syntax: show ip pim resource

For each software data structure listed in the output, the following information is shown:

Table 30.4: Output of show ip pim resource

This Field...	Displays...
alloc	Number of nodes of that data that are currently allocated in memory.
in-use	Number of allocated nodes in use
avail	Number of allocated nodes are not in use
allo-fail	Number of allocated notes that failed

Table 30.4: Output of show ip pim resource (Continued)

This Field...	Displays...
up-limit	Maximum number of nodes that can be allocated for a data structure. This may or may not be configurable, depending on the data structure
get-mem	Number of attempts made to use allocated nodes
#of PIM ports	Total number of PIM ports, by port type, on the device
Total, allocated, and available Mils	In Layer 3 multicast, this refers to the Multicast Linked List that contains information on where (S,G) gets forwarded. Each (S,G) entry requires a single MLL entry to forward traffic to all physical, untagged ports. Also, one MLL entry is required per VLAN that has tagged outbound ports. There can be up to 1024 MLL entries.

Displaying Candidate RP Information

To display candidate RP information, enter the following command at any CLI level:

```
FastIron#show ip pim rp-candidate

Next Candidate-RP-advertisement in 00:00:10
  RP: 207.95.7.1
    group prefixes:
    224.0.0.0 / 4

  Candidate-RP-advertisement period: 60
```

This example show information displayed on a Layer 3 Switch that is a candidate RP. The following example shows the message displayed on a Layer 3 Switch that is not a candidate RP.

```
FastIron#show ip pim rp-candidate
```

This system is not a Candidate-RP.

Syntax: show ip pim rp-candidate

This display shows the following information.

Table 30.5: Output of show ip pim rp-candidate

This Field...	Displays...
Candidate-RP-advertisement in	Indicates how many seconds will pass before the BSR sends its next RP message. **Note**: This field appears only if this Layer 3 Switch is a candidate RP.
RP	Indicates the IP address of the Rendezvous Point (RP). **Note**: This field appears only if this Layer 3 Switch is a candidate RP.
group prefixes	Indicates the multicast groups for which the RP listed by the previous field is a candidate RP. **Note**: This field appears only if this Layer 3 Switch is a candidate RP.

Table 30.5: Output of show ip pim rp-candidate (Continued)

This Field...	Displays...
Candidate-RP-advertisement period	Indicates how frequently the BSR sends candidate RP advertisement messages. **Note**: This field appears only if this Layer 3 Switch is a candidate RP.

Displaying RP-to-Group Mappings

To display RP-to-group-mappings, enter the following command at any CLI level:

```
FastIron#show ip pim rp-map
Number of group-to-RP mappings: 6

Group address        RP address
-------------------------------
1 239.255.163.1  99.99.99.5
2 239.255.163.2  99.99.99.5
3 239.255.163.3  99.99.99.5
4 239.255.162.1  99.99.99.5
5 239.255.162.2  43.43.43.1
6 239.255.162.3  99.99.99.5
```

Syntax: show ip pim rp-map

This display shows the following information.

Table 30.6: Output of show ip pim rp-map

This Field...	Displays...
Group address	Indicates the PIM Sparse multicast group address using the listed RP.
RP address	Indicates the IP address of the Rendezvous Point (RP) for the listed PIM Sparse group.

Displaying RP Information for a PIM Sparse Group

To display RP information for a PIM Sparse group, enter the following command at any CLI level:

```
 FastIron#show ip pim rp-hash 239.255.162.1

   RP: 207.95.7.1, v2
     Info source: 207.95.7.1, via bootstrap
```

Syntax: show ip pim rp-hash <group-addr>

The <group-addr> parameter is the address of a PIM Sparse IP multicast group.

This display shows the following information.

Table 30.7: Output of show ip pim rp-hash

This Field...	Displays...
RP	Indicates the IP address of the Rendezvous Point (RP) for the specified PIM Sparse group. Following the IP address is the port or virtual interface through which this Layer 3 Switch learned the identity of the RP.
Info source	Indicates the IP address on which the RP information was received. Following the IP address is the method through which this Layer 3 Switch learned the identity of the RP.

Displaying the RP Set List

To display the RP set list, enter the following command at any CLI level:

```
FastIron#show ip pim rp-set
Group address Static-RP-address Override
---------------------------------------------------
Access-List 44 99.99.99.5 On
Number of group prefixes Learnt from BSR: 1
Group prefix = 239.255.162.0/24 #RPs expected: 1
#RPs received: 1
RP 1: 43.43.43.1 priority=0 age=0
```

Syntax: show ip pim rp-set

This display shows the following information.

Table 30.8: Output of show ip pim rp-set

This Field...	Displays...
Number of group prefixes	The number f PIM Sparse group prefixes for which the RP is responsible.
Group prefix	Indicates the multicast groups for which the RP listed by the previous field is a candidate RP.
RPs expected/received	Indicates how many RPs were expected and received in the latest Bootstrap message.
RP <num>	Indicates the RP number. If there are multiple RPs in the PIM Sparse domain, a line of information for each of them is listed, and they are numbered in ascending numerical order.
priority	The RP priority of the candidate RP. During the election process, the candidate RP with the highest priority is elected as the RP.
age	The age (in seconds) of this RP-set. **Note**: If this Layer 3 Switch is not a BSR, this field contains zero. Only the BSR ages the RP-set.

Displaying Multicast Neighbor Information

To display information about the Layer 3 Switch's PIM neighbors, enter the following command at any CLI level:

```
FastIron#show ip pim nbr

Port Neighbor        Holdtime Age   UpTime
                     sec      sec   sec
e3/8   207.95.8.10     180      60    900
Port Neighbor        Holdtime Age   UpTime
                     sec      sec   sec
v1     207.95.6.2      180      60    900
```

Syntax: show ip pim nbr

This display shows the following information.

Table 30.9: Output of show ip pim nbr

This Field...	Displays...
Port	The interface through which the Layer 3 Switch is connected to the neighbor.
Neighbor	The IP interface of the PIM neighbor interface.
Holdtime sec	Indicates how many seconds the neighbor wants this Layer 3 Switch to hold the entry for this neighbor in memory. The neighbor sends the Hold Time in its Hello packets. • If the Layer 3 Switch receives a new Hello packet before the Hold Time received in the previous packet expires, the Layer 3 Switch updates its table entry for the neighbor. • If the Layer 3 Switch does not receive a new Hello packet from the neighbor before the Hold time expires, the Layer 3 Switch assumes the neighbor is no longer available and removes the entry for the neighbor.
Age sec	The number of seconds since the Layer 3 Switch received the last hello message from the neighbor.
UpTime sec	The number of seconds the PIM neighbor has been up. This timer starts when the Layer 3 Switch receives the first Hello messages from the neighbor.

Displaying Information About an Upstream Neighbor Device

You can view information about the upstream neighbor device for a given source IP address for IP Protocol Independent Multicast (PIM) and Distance Vector Multicast Routing Protocol (DVMRP) packets. For PIM, the software uses the IP route table or multicast route table to lookup the upstream neighbor device. For DVMRP, the software uses the DVMRP route table to locate the upstream neighbor device.

Enter the following command at the Privileged EXEC level of the CLI:

```
FastIron#show ip pim rpf 1.1.20.2
```

```
directly connected or via an L2 neighbor
```

Syntax: show ip pim | dvmrp rpf <IP address>

where <IP address> is a valid source IP address

NOTE: If there are multiple equal cost paths to the source, the **show ip pim rpf** command output may not be accurate. If your system has multiple equal cost paths, use the command **sh ip pim mcache** to view information about the upstream neighbor.

Displaying the PIM Flow Cache

To display the PIM flow cache, enter the following command at any CLI level:

```
FastIron#show ip pim flowcache

     Source          Group           Parent CamFlags CamIndex  Fid      Flags
1    209.157.24.162  239.255.162.1   v2     00000700 2023      00004411 F
2    209.157.24.162  239.255.162.1   v2     00000700 201b      00004411 F
3    209.157.24.162  239.255.162.1   v2     00000700 201d      00004411 F
4    209.157.24.162  239.255.162.1   v2     00000700 201e      00004411 F
```

Syntax: show ip pim flowcache

This display shows the following information.

Table 30.10: Output of show ip pim flowcache

This Field...	Displays...
Source	Indicates the source of the PIM Sparse group.
Group	Indicates the PIM Sparse group.
Parent	Indicates the port or virtual interface from which the Layer 3 Switch receives packets from the group’s source.
CamFlags	This field is used by Foundry technical support for troubleshooting.
CamIndex	This field is used by Foundry technical support for troubleshooting.
Fid	This field is used by Foundry technical support for troubleshooting.
Flags	This field is used by Foundry technical support for troubleshooting.

Displaying the PIM Multicast Cache

To display the PIM multicast cache, enter the following command at any CLI level:

```
FastIron#show ip pim mcache

1    (*,239.255.162.1) RP207.95.7.1 forward port v1, Count 2
     member ports ethe 3/3
     virtual ports v2
     prune ports
     virtual prune ports

2    (209.157.24.162,239.255.162.4) forward port v2, flags 00004900 Count 130
     member ports
     virtual ports
     prune ports
     virtual prune ports

3    (209.157.24.162,239.255.162.1) forward port v2, flags 00005a01 Count 12
     member ports ethe 3/8
     virtual ports
     prune ports
     virtual prune ports
```

Syntax: show ip pim mcache

This display shows the following information.

Table 30.11: Output of show ip pim mcache

This Field...	Displays...
(*<source>*, *<group>*)	The comma-separated values in parentheses is a source-group pair. The *<source>* is the PIM source for the multicast *<group>*. For example, the following entry means source 209.157.24.162 for group 239.255.162.1: (209.157.24.162,239.255.162.1) If the *<source>* value is * (asterisk), this cache entry uses the RP path. The * value means “all sources”. If the *<source>* is a specific source address, this cache entry uses the SPT path.
RP<ip-addr>	Indicates the RP for the group for this cache entry. **Note**: The RP address appears only if the RPT flag is set to 1 and the SPT flag is set to 0 (see below).
forward port	The port through which the Layer 3 Switch reaches the source.
Count	The number of packets forwarded using this cache entry.
Sparse Mode	Indicates whether the cache entry is for regular PIM (dense mode) or PIM Sparse. This flag can have one of the following values: • 0 – The entry is not for PIM Sparse (and is therefore for the dense mode of PIM). • 1– The entry is for PIM Sparse.

Table 30.11: Output of show ip pim mcache (Continued)

This Field...	Displays...
RPT	Indicates whether the cache entry uses the RP path or the SPT path. The RPT flag can have one of the following values: • 0 – The SPT path is used instead of the RP path. • 1– The RP path is used instead of the SPT path. **Note**: The values of the RP and SPT flags are always opposite (one is set to 0 and the other is set to 1).
SPT	Indicates whether the cache entry uses the RP path or the SPT path. The SP flag can have one of the following values: • 0 – The RP path is used instead of the SPT path. • 1– The SPT path is used instead of the RP path. **Note**: The values of the RP and SPT flags are always opposite (one is set to 0 and the other is set to 1).
Register Suppress	Indicates whether the Register Suppress timer is running. This field can have one of the following values: • 0 – The timer is not running. • 1 – The timer is running.
member ports	Indicates the Layer 3 Switch physical ports to which the receivers for the source and group are attached. The receivers can be directly attached or indirectly attached through other PIM Sparse routers.
virtual ports	Indicates the virtual interfaces to which the receivers for the source and group are attached. The receivers can be directly attached or indirectly attached through other PIM Sparse routers.
prune ports	Indicates the physical ports on which the Layer 3 Switch has received a prune notification (in a Join/Prune message) to remove the receiver from the list of recipients for the group.
virtual prune ports	Indicates the virtual interfaces ports on which the Layer 3 Switch has received a prune notification (in a Join/Prune message) to remove the receiver from the list of recipients for the group.

Displaying PIM Traffic Statistics

To display PIM traffic statistics, use the following CLI method.

```
FastIron#show ip pim traffic

Port    Hello          J/P          Register      RegStop          Assert
     [Rx     Tx]    [Rx     Tx]    [Rx     Tx]    [Rx     Tx]    [Rx     Tx]
e3/8   19     19      32     0      0      0      37     0      0      0

Port    Hello          J/P          Register      RegStop          Assert
     [Rx     Tx]    [Rx     Tx]    [Rx     Tx]    [Rx     Tx]    [Rx     Tx]
v1     18     19      0      20     0      0      0      0      0      0

Port    Hello          J/P          Register      RegStop          Assert
     [Rx     Tx]    [Rx     Tx]    [Rx     Tx]    [Rx     Tx]    [Rx     Tx]
v2     0      19      0      0      0      16     0      0      0      0

Total 37      57     32      0      0      0      0      0      0      0
IGMP Statistics:
  Total Recv/Xmit 85/110
  Total Discard/chksum  0/0
```

Syntax: show ip pim traffic

NOTE: If you have configured interfaces for standard PIM (dense mode) on the Layer 3 Switch, statistics for these interfaces are listed first by the display.

This display shows the following information.

Table 30.12: Output of show ip pim traffic

This Field...	Displays...
Port	The port or virtual interface on which the PIM interface is configured.
Hello	The number of PIM Hello messages sent or received on the interface.
J/P	The number of Join/Prune messages sent or received on the interface. **Note**: Unlike PIM dense, PIM Sparse uses the same messages for Joins and Prunes.
Register	The number of Register messages sent or received on the interface.
RegStop	The number of Register Stop messages sent or received on the interface.
Assert	The number of Assert messages sent or received on the interface.
Total Recv/Xmit	The total number of IGMP messages sent and received by the Layer 3 Switch.
Total Discard/chksum	The total number of IGMP messages discarded, including a separate counter for those that failed the checksum comparison.

Displaying and Clearing PIM Errors

If you want to determine how many PIM errors there are on the device, enter the following command:

```
FastIron#show ip pim error
**** Warning counter pim route change = 1
HW tagged replication enabled, SW processed pkts 0
```

Syntax: show ip pim error

This command displays the number of warnings and non-zero PIM errors on the device. This count can increase during transition periods such as reboots and topology changes; however, if the device is stable, the number of errors should not increase. If warnings keep increasing in a stable topology, then there may be a configuration error or problems on the device.

To clear the counter for PIM errors, enter the following command:

```
FastIron#clear pim counters
```

Syntax: clear pim counters

Passive Multicast Route Insertion

Platform Support:

- FESX/FSX/FWSX devices running software release 02.4.00 and later

Passive Multicast Route Insertion (PMRI) enables a Layer 3 switch running PIM Sparse to create an entry for a multicast route (e.g., (S,G)), with no directly attached clients or when connected to another PIM router (transit network).

PMRI is critical for Service Providers wanting to deliver IP-TV services or multicast-based video services. Service Providers, who have transit networks, distribute multicast-based video services to other Service Providers, regardless of whether a client subscribes to a video service.

To configure PMRI, enter the following command at the **router pim** level of the CLI:

```
FastIron(config)#router pim
FastIron#(config-pim-router)#hardware-drop
```

Syntax: [no] hardware-drop

When you enable PMRI, the **show ip pim mcache** command output displays the multicast cache entry along with a **drop** flag, indicating that the device is dropping packets in hardware. If the **HW** flag is set to 1 (**HW=1**), it implies that the packets are being dropped in hardware. If the **HW** flag is set to 0, (**HW=0)**, it indicates that the packets are being processed in software. The following shows an example display output.

```
FastIron#show ip pim mcache
1 (10.10.10.18 226.0.1.56) in v10 (e1), cnt=2
Source is directly connected
Sparse Mode, RPT=0 SPT=1 REG=1 MSDP Adv=0 MSDP Create=0
fast=0 slow=0 pru=1 graft age drop
age=0s up-time=2m HW=1 L2-vidx=8191
```

DVMRP Overview

Foundry routers provide multicast routing with the ***Distance Vector Multicast Routing Protocol (DVMRP)*** routing protocol. DVMRP uses ***Internet Group Membership Protocol (IGMP)*** to manage the IP multicast groups.

DVMRP is a broadcast and pruning multicast protocol that delivers IP multicast datagrams to its intended receivers. The receiver registers the interested groups using IGMP. DVMRP builds a multicast delivery tree with the sender forming the root. Initially, multicast datagrams are delivered to all nodes on the tree. Those leaves that do not have any group members send ***prune messages*** to the upstream router, noting the absence of a group.

The upstream router maintains a prune state for this group for the given sender. A prune state is aged out after a given configurable interval, allowing multicasts to resume.

DVMRP employs ***reverse path forwarding*** and ***pruning*** to keep source specific multicast delivery trees with the minimum number of branches required to reach all group members. DVMRP builds a multicast tree for each source and destination host group.

Initiating DVMRP Multicasts on a Network

Once DVMRP is enabled on each router, a network user can begin a video conference multicast from the server on R1. ***Multicast Delivery Trees*** are initially formed by source-originated multicast packets that are propagated to downstream interfaces as seen in Figure 30.4. When a multicast packet is received on a DVMRP-capable router interface, the interface checks its DVMRP routing table to determine whether the interface that received the message provides the shortest path back to the source. If the interface does provide the shortest path, the interface forwards the multicast packet to adjacent peer DVMRP routers, except for the router interface that originated the packet. Otherwise, the interface discards the multicast packet and sends a prune message back upstream. This process is known as ***reverse path forwarding***.

In Figure 30.4, the root node (R1) is forwarding multicast packets for group 229.225.0.2 that it receives from the server to its downstream nodes, R2, R3, and R4. Router R4 is an intermediate router with R5 and R6 as its downstream routers. Because R5 and R6 have no downstream interfaces, they are leaf nodes.

The receivers in this example are those workstations that are resident on routers R2, R3, and R6.

Pruning a Multicast Tree

After the multicast tree is constructed, ***pruning*** of the tree will occur after IP multicast packets begin to traverse the tree.

As multicast packets reach leaf networks (subnets with no downstream interfaces), the local IGMP database checks for the recently arrived IP multicast packet address. If the local database does not contain the address (the address has not been learned), the router prunes (removes) the address from the multicast tree and no longer receives multicasts until the prune age expires.

In Figure 30.5, Router 5 is a leaf node with no group members in its local database. Consequently, Router 5 sends a prune message to its upstream router. This router will not receive any further multicast traffic until the prune age interval expires.

Figure 30.4 Downstream Broadcast of IP Multicast Packets from Source Host

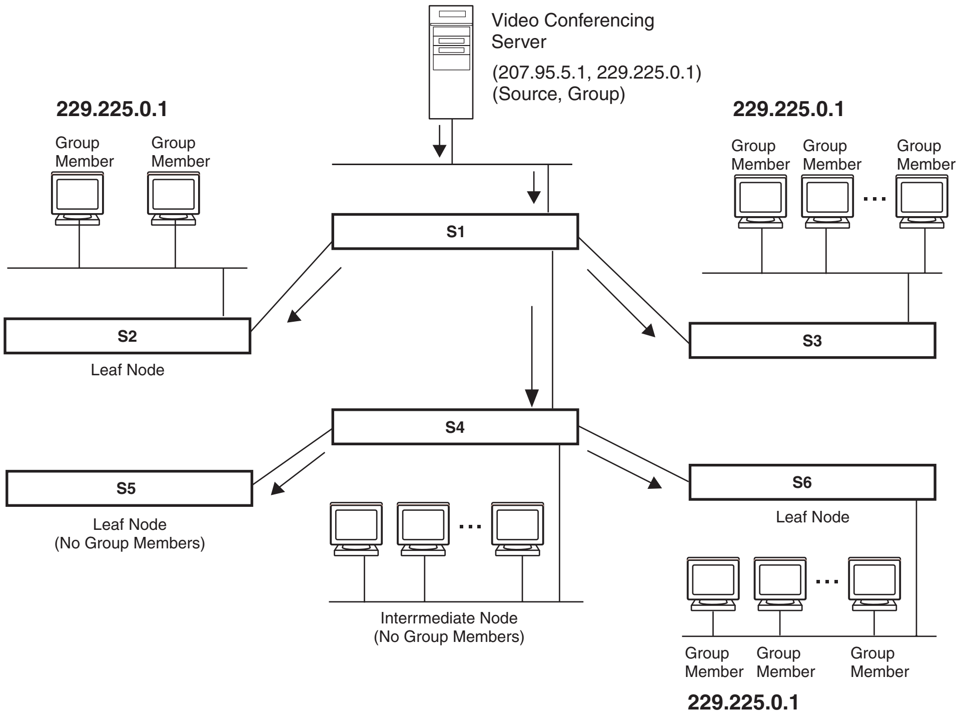
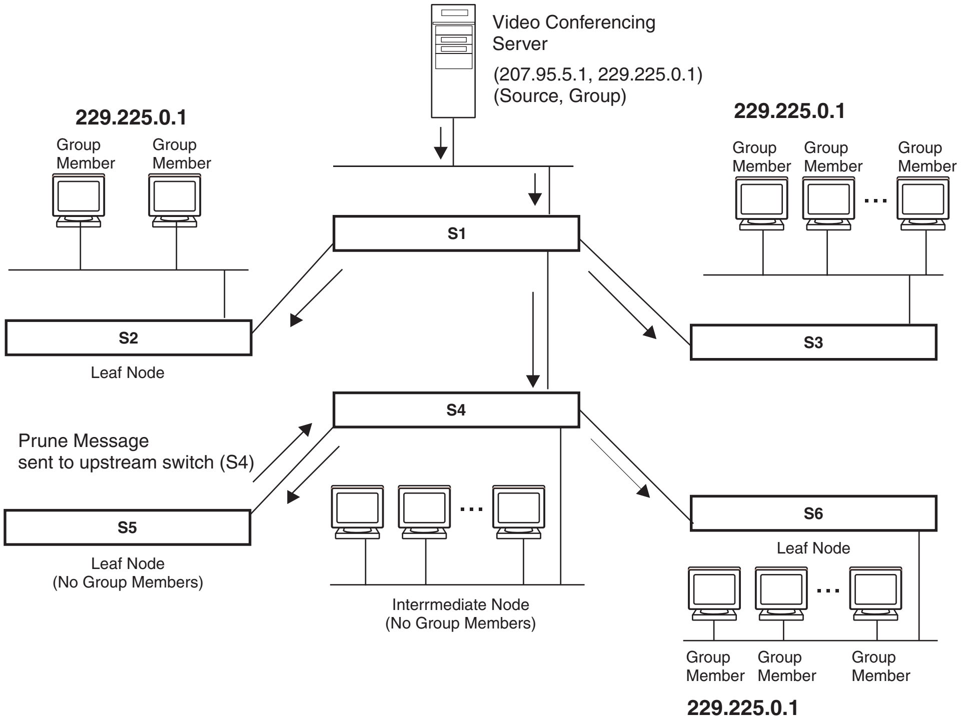


Figure 30.5 Pruning Leaf Nodes from a Multicast Tree



Grafts to a Multicast Tree

A DVMRP router restores pruned branches to a multicast tree by sending graft messages towards the upstream switch. Graft messages start at the leaf node and travel up the tree, first sending the message to its neighbor upstream switch.

In the example above, if a new 229.255.0.1 group member joins on switch S6, which had been pruned previously, a graft will be sent upstream to S4. Since the forwarding state for this entry is in a prune state, S4 sends a graft to S1. Once S4 has joined the tree, it and S6 will once again receive multicast packets.

You do not need to perform any configuration to maintain the multicast delivery tree. The prune and graft messages automatically maintain the tree.

Configuring DVMRP

Enabling DVMRP on the Layer 3 Switch and Interface

Suppose you want to initiate the use of desktop video for fellow users on a sprawling campus network. All destination workstations have the appropriate hardware and software but the Layer 3 Switches that connect the various buildings need to be configured to support DVMRP multicasts from the designated video conference server as seen in Figure 30.4.

DVMRP is enabled on each of the Foundry Layer 3 Switches shown in Figure 30.4, on which multicasts are expected. You can enable DVMRP on each Layer 3 Switch independently or remotely from one Layer 3 Switch by a Telnet connection. Follow the same steps for each Layer 3 Switch.

Globally Enabling and Disabling DVMRP

To globally enable DVMRP, enter the following command:

```
Router1(config)#router dvmrp
```

Syntax: [no] router dvmrp

The behavior of the **[no] router dvmrp** command is as follows:

- Entering a **router dvmrp** command to enable DVMRP does not require a software reload.
- Entering a **no router dvmrp** command removes all configuration for PIM multicast on a Layer 3 Switch (**router pim** level) only.

Globally Enabling or Disabling DVMRP without Deleting Multicast Configuration

As stated above enter **no router dvmrp** removed PIM configuration. If you want to disable or enable DVMRP without removing PIM configuration, enter the following command:

```
FastIron(config)#router dvmrp
FastIron(config-pim-router)#disable-dvmrp
```

Syntax: [no] disable-dvmrp

Use the [no] version of the command to re-enable DVMRP.

Enabling DVMRP on an Interface

After globally enabling DVMRP on a Layer 3 Switch, enable it on each interface that will support the protocol.

To enable DVMRP on S1 and interface 3, enter the following:

```
Router1(config)#router dvmrp
Router1(config-dvmrp-router)#int e 3
Router1(config-if-3)#ip dvmrp
```

Modifying DVMRP Global Parameters

DVMRP global parameters come with preset values. The defaults work well in most networks, but you can modify the following global parameters if you need to:

- Neighbor timeout
- Route expire time
- Route discard time
- Prune age
- Graft retransmit time
- Probe interval
- Report interval
- Trigger interval
- Default route

Modifying Neighbor Timeout

The neighbor timeout specifies the period of time that a router will wait before it defines an attached DVMRP neighbor router as down. Possible values are 40 – 8000 seconds. The default value is 180 seconds.

To modify the neighbor timeout value to 100, enter the following:

```
FastIron(config-dvmrp-router)#nbr 100
```

Syntax: nbr-timeout <40-8000>

The default is 180 seconds.

Modifying Route Expires Time

The Route Expire Time defines how long a route is considered valid in the absence of the next route update. Possible values are from 20 – 4000 seconds. The default value is 200 seconds.

To modify the route expire setting to 50, enter the following:

```
FastIron(config-dvmrp-router)#route-expire-timeout 50
```

Syntax: route-expire-timeout <20-4000>

Modifying Route Discard Time

The Route Discard Time defines the period of time before a route is deleted. Possible values are from 40 – 8000 seconds. The default value is 340 seconds.

To modify the route discard setting to 150, enter the following:

```
FastIron(config-dvmrp-router)#route-discard-timeout 150
```

Syntax: route-discard-timeout <40-8000>

Modifying Prune Age

The Prune Age defines how long a prune state will remain in effect for a source-routed multicast tree. After the prune age period expires, flooding will resume. Possible values are from 20 – 3600 seconds. The default value is 180 seconds.

To modify the prune age setting to 150, enter the following:

```
FastIron(config-dvmrp-router)#prune 25
```

Syntax: prune-age <20-3600>

Modifying Graft Retransmit Time

The Graft Retransmit Time defines the initial period of time that a router sending a graft message will wait for a graft acknowledgement from an upstream router before re-transmitting that message.

Subsequent retransmissions are sent at an interval twice that of the preceding interval. Possible values are from 5 – 3600 seconds. The default value is 10 seconds.

To modify the setting for graft retransmit time to 120, enter the following:

```
FastIron(config-dvmrp-router)#graft 120
```

Syntax: graft-retransmit-time <5-3600>

Modifying Probe Interval

The Probe Interval defines how often neighbor probe messages are sent to the ALL-DVMRP-ROUTERS IP multicast group address. A router's probe message lists those neighbor DVMRP routers from which it has received probes. Possible values are from 5 – 30 seconds. The default value is 10 seconds.

To modify the probe interval setting to 10, enter the following:

```
FastIron(config-dvmrp-router)#probe 10
```

Syntax: probe-interval <5-30>

Modifying Report Interval

The Report Interval defines how often routers propagate their complete routing tables to other neighbor DVMRP routers. Possible values are from 10 – 2000 seconds. The default value is 60 seconds.

To support propagation of DVMRP routing information to the network every 90 seconds, enter the following:

```
FastIron(config-dvmrp-router)#report 90
```

Syntax: report-interval <10-2000>

Modifying Trigger Interval

The Trigger Interval defines how often trigger updates, which reflect changes in the network topology, are sent. Example changes in a network topology include router up or down or changes in the metric. Possible values are from 5 – 30 seconds. The default value is 5 seconds.

To support the sending of trigger updates every 20 seconds, enter the following:

```
FastIron(config-dvmrp-router)#trigger-interval 20
```

Syntax: trigger-interval <5-30>

Modifying Default Route

To define the default gateway for DVMRP, enter the following:

```
FastIron(config-dvmrp-router)#default-gateway 192.35.4.1
```

Syntax: default-gateway <ip-addr>

Modifying DVMRP Interface Parameters

DVMRP global parameters come with preset values. The defaults work well in most networks, but you can modify the following interface parameters if you need to:

- TTL
- Metric
- Advertising

Modifying the TTL

The TTL defines the minimum value required in a packet in order for the packet to be forwarded out the interface. For example, if the TTL for an interface is set at 10 it means that only those packets with a TTL value of 10 or more are forwarded. Likewise, if an interface is configured with a TTL Threshold value of 1, all packets received on that interface are forwarded. Possible values are from 1 – 64. The default value is 1.

To set a TTL of 64, enter the following:

```
FastIron(config)#int e 1/4
FastIron(config-if-1/4)#ip dvmrp ttl 60
```

Syntax: ttl-threshold <1-64>

Modifying the Metric

The router uses the metric when establishing reverse paths to some networks on directly attached interfaces. Possible values are from 1 – 31 hops. The default is 1.

NOTE: This command is not supported on Foundry Layer 2 Switches.

To set a metric of 15 for a DVMRP interface, enter the following:

```
FastIron(config)#interface 3/5
FastIron(config-if-3/5)#ip dvmrp metric 15
```

Syntax: ip dvmrp metric <1-31>

Enabling Advertising

You can turn the advertisement of a local route on (enable) or off (disable) on the interface. By default, advertising is enabled.

To enable advertising on an interface, enter the following:

```
FastIron(config-if-1/4)#ip dvmrp advertise-local on
```

Syntax: advertise-local on | off

Displaying Information About an Upstream Neighbor Device

You can view information about the upstream neighbor device for a given source IP address for IP PIM packets. The software uses the IP route table or multicast route table to lookup the upstream neighbor device.

The following shows example messages that the Foundry device can display with this command.

```
FastIron#show ip dvmrp rpf 1.1.20.2
directly connected or via an L2 neighbor
FastIron#show ip dvmrp rpf 1.2.3.4
no route
FastIron#show ip dvmrp rpf 1.10.10.24
upstream neighbor=1.1.20.1 on v21 using ip route
```

Syntax: show ip dvmrp rpf <IP address>

where <IP address> is a valid source IP address

NOTE: If there are multiple equal cost paths to the source, the **show ip dvmrp rpf** command output may not be accurate. If your system has multiple equal cost paths, use the command **sh ip dvmrp mcache** to view information about the upstream neighbor.

Configuring an IP Tunnel

IP tunnels are used to send traffic through routers that do not support IP multicasting. IP Multicast datagrams are encapsulated within an IP packet and then sent to the remote address. Routers that are not configured for IP Multicast route that packet as a normal IP packet. When the IP Multicast router at the remote end of the tunnel receives the packet, the router strips off the IP encapsulation and forwards the packet as an IP Multicast packet.

NOTE: An IP tunnel must have a remote IP interface at each end. Also, for IP tunneling to work, the remote routers must be reachable by an IP routing protocol.

NOTE: Multiple tunnels configured on a router cannot share the same remote address.

EXAMPLES:

To configure an IP tunnel as seen in Figure 30.6, enter the IP tunnel destination address on an interface of the router.

To configure an IP address on Router A, enter the following:

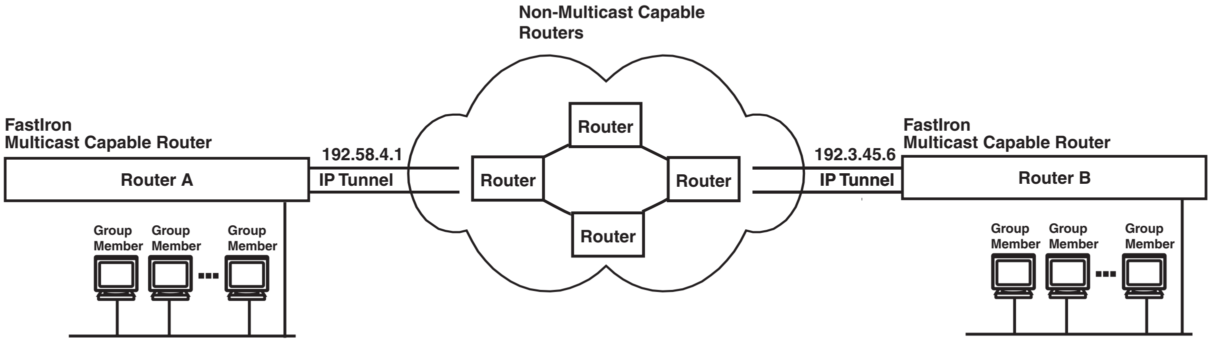
```
FastIron(config)#int e1
FastIron(config-if-1)#ip tunnel 192.3.45.6
```

NOTE: The IP tunnel address represents the configured IP tunnel address of the destination router. In the case of Router A, its destination router is Router B. Router A is the destination router of Router B.

For router B, enter the following:

```
FastIron(config-if-1)#ip tunnel 192.58.4.1
```

Figure 30.6 IP in IP Tunneling on Multicast Packets in a Unicast Network



Using ACLs to Control Multicast Features

You can use ACLs to control the following multicast features:

- Limit the number of multicast groups that are covered by a static rendezvous point (RP)
- Control which multicast groups for which candidate RPs sends advertisement messages to bootstrap routers
- Identify which multicast group packets will be forwarded or blocked on an interface

Using ACLs to Limit Static RP Groups

You can limit the number of multicast groups covered by a static RP using standard ACLs. In the ACL, you specify the group to which the RP address applies. The following examples set the RP address to be applied to multicast groups with some minor variations.

To configure an RP that covers multicast groups in 239.255.162.x, enter commands such as the following:

```
FastIron(config)#access-list 2 permit 239.255.162.0 0.0.0.255

FastIron(config)#router pim
FastIron(config-pim-router)#rp-address 43.43.43.1 2
```

To configure an RP that covers multicast groups in the 239.255.162.x range, except the 239.255.162.2 group, enter commands such as the following:

```
FastIron(config)#access-list 5 deny host 239.255.162.2
FastIron(config)#access-list 5 permit 239.255.0.0 0.0.255.255

FastIron(config)#router pim
FastIron(config-pim-router)#bsr-candidate ve 43 32 100
FastIron(config-pim-router)#rp-candidate ve 43
FastIron(config-pim-router)#rp-address 99.99.99.5 5
```

To configure an RP for multicast groups using the override switch, enter commands such as the following:

```
FastIron(config)#access-list 44 permit 239.255.162.0 0.0.0.255

FastIron(config)#router pim
FastIron(config-pim-router)#rp-address 43.43.43.1
FastIron(config-pim-router)#rp-address 99.99.99.5 44 override
```

Syntax: [no] rp-address <ip-address> [<access-list-num>] [override]

The access-list-num parameter is the number of the standard ACL that will filter the multicast group.

NOTE: Extended ACLs cannot be used to limit static RP groups.

The **override** parameter directs the Layer 3 Switch to ignore the information learned by a BSR if there is a conflict between the RP configured in this command and the information that is learned by the BSR. In previous releases, static RP configuration precedes the RP address learned from the PIM Bootstrap protocol. With this enhancement, an RP address learned dynamically from PIM Bootstrap protocol takes precedence over static RP configuration unless the override parameter is used.

You can use the **show ip pim rp-set** command to display the ACLs used to filter the static RP groups. For example,

```
FastIron#show ip pim rp-set

Group address    Static-RP-address  Override
---------------------------------------------------
Access-List 44    99.99.99.5           On

Number of group prefixes Learnt from BSR: 1

Group prefix = 224.0.0.0/4 #RPs: 1
      RP 1: 43.43.43.1 priority=0 age=0
```

In the example above, the display shows the following information:

- The Group Address table shows the static RP address that is covered by the access list, and whether or not the override parameter has been enabled.
- The Group prefix line shows the multicast group prefix for the static RP.
- The RP #line shows the configured IP address of the RP candidate.

The **show ip pim rp-map** to show the group-to-RP mapping.

```
FastIron#show ip pim rp-map

Number of group-to-RP mappings: 6

  Group address  RP address
-------------------------------
1 239.255.163.1  43.43.43.1
2 239.255.163.2  43.43.43.1
3 239.255.163.3  43.43.43.1
4 239.255.162.1  99.99.99.5
5 239.255.162.2  99.99.99.5
6 239.255.162.3  99.99.99.5
```

The display shows the multicast group addresses covered by the RP candidate and the IP address of the RP for the listed multicast group. In the example above, you see the following:

- The first three lines show the multicast group addresses that are covered by the RP candidate.
- The last three lines show the multicast group addresses covered by the static RP.

Using ACLs to Limit PIM RP Candidate Advertisement

You can use standard ACLs to control the groups for which the candidate RP will send advertisement messages to the bootstrap router. For example, ACL 5 can be configured to be applied to the multicast groups within the IP address 239.x.x.x range. You can configure the Layer 3 Switch to advertise itself as a candidate RP to the bootstrap router only for groups in the range of 239.x.x.x. Enter commands such as the following:

```
FastIron(config)#interface ethernet 1/1
FastIron(config-if-1/1)#ip address 99.99.99.5 255.255.255.0
FastIron(config-if-1/1)#ip pim-sparse
FastIron(config-if-1/1)#exit

FastIron(config)#access-list 5 deny host 239.255.162.2
FastIron(config)#access-list 5 permit 239.0.0.0 0.0.255.255

FastIron(config)#router pim
FastIron(config-pim-router)#bsr-candidate ethernet 1/1 32 100
FastIron(config-pim-router)#rp-candidate ethernet 1/1 group-list 5
```

The example above shows a configuration for an Ethernet interface. To configure ACLs that are applied to a virtual routing interface, enter commands such as the following:

```
FastIron(config)#interface ve 16
FastIron(config-vif-16)#ip address 16.16.16.1 255.255.255.0
FastIron(config-vif-16)#ip pim-sparse
FastIron(config-vif-16)#exit

FastIron(config)#access-list 5 deny host 239.255.162.2
FastIron(config)#access-list 5 permit 239.255.0.0 0.0.255.255

FastIron(config)#router pim
FastIron(config-pim-router)#bsr-candidate ve 16 32 100
FastIron(config-pim-router)#rp-candidate ve 16 group-list 5
```

To configure ACLs that are applied to a loopback interface, enter commands such as the following:

```
FastIron(config)#interface loopback 1
FastIron(config-lbif-1)#ip address 88.88.88.8 255.255.255.0
FastIron(config-lbif-1)#ip pim-sparse
FastIron(config-lbif-1)#exit

FastIron(config)#access-list 5 deny host 239.255.162.2
FastIron(config)#access-list 5 permit 239.255.0.0 0.0.255.255
```

```
FastIron(config)#router pim
FastIron(config-pim-router)#bsr-candidate loopback 1 32 100
FastIron(config-pim-router)#rp-candidate loopback 1 group-list 5
```

Syntax: [no] rp-candidate ethernet [<slotnum>/]<portnum> | loopback <num> | ve <num> [group-list <access-list-num>]

The <slotnum> parameter is required on chassis devices.

The <portnum> | **loopback** <num> | **ve** <num> parameter specifies the interface. The Layer 3 Switch will advertise the specified interface's IP address as a candidate RP.

- Enter **ethernet** [<slotnum>/]<portnum> for a physical interface (port).
- Enter **ve** <num> for a virtual interface.
- Enter **loopback** <num> for a loopback interface.

The **group-list** <access-list-num> indicates that a standard ACL is used to filter for which multicast group the advertisement will be made.

NOTE: Extended ACLs cannot be used for group-list.

Disabling CPU Processing for Selective Multicast Groups

Platform Support:

- FESX and FSX devices running software release 04.1.00 and later – L2, BL3, L3

In IPv4 multicast, Foundry Layer 3 switches do not forward multicast packets with destination addresses in the range between 224.0.0.0 and 224.0.0.255. These group addresses are reserved for various routing protocols. By default, packets destined to these groups are processed by the CPU. However, when a large number of packets for these groups are received by the Foundry device at the same time, the CPU could get overwhelmed. To alleviate the load on the CPU, you could disable CPU processing of packets for these groups. When applied, this feature protects the CPU from traffic sent to IPV4 multicast addresses in the range 224.0.0.1 - 224.0.0.254, and instead floods these packets in hardware within the incoming VLAN.

This feature can be applied on a VLAN or a VLAN-group. If applied on a VLAN, traffic received on a port of the VLAN will be flooded to all other ports of the VLAN. If applied on a VLAN-group, traffic will be flooded only at the individual VLAN level. Once this feature is applied on a VLAN or VLAN-group, ports that are statically or dynamically added to the VLAN or VLAN-group will inherit the configuration. Likewise, ports that are statically or dynamically removed from the VLAN or VLAN-group will drop the configuration.

This feature can be enabled for packets destined to a multicast group or set of groups in the range 224.0.0.1 – 224.0.0.254, except for the reserved multicast addresses listed in the following table.

Table 30.13: Reserved Multicast Addresses

Multicast Address	Reserved for...
224.0.0.1	all nodes
224.0.0.2	PIM
224.0.0.3	DVMRP
224.0.0.4	DVMRP
224.0.0.5	OSPF
224.0.0.6	OSPF

Table 30.13: Reserved Multicast Addresses (Continued)

Multicast Address	Reserved for...
224.0.0.9	RIP V2
224.0.0.13	PIM V2
224.0.0.18	VRRP
224.0.0.22	IGMP V3 reports

CLI Command Syntax

To disable CPU processing for selective multicast groups, enter commands such as the following:

```
FastIron# config t
FastIron(config)# vlan 5
FastIron(config-vlan-5)# disable multicast-to-cpu 224.0.0.5
FastIron(config-vlan-5)# disable multicast-to-cpu 224.0.0.14 224.0.0.230
FastIron(config-vlan-5)# vlan 10
FastIron(config-vlan-10)# disable multicast-to-cpu 224.0.0.23
FastIron(config-vlan-10)# vlan 20
FastIron(config-vlan-20)# disable multicast-to-cpu 224.0.0.50 224.0.0.140
```

Syntax: [no] disable multicast-to-cpu <multicast group address> [<multicast group range end address>]

The <multicast group address> must be in the range 224.0.0.1 - 224.0.0.254, but cannot be one of the reserved multicast addresses listed in Table 30.13 on page 30-42.

Viewing Disabled Multicast Addresses

To display disabled multicast addresses for all configured VLANs, enter the command **show disabled-multicast-to-cpu**. The following shows an example display.

```
Router# show disabled-multicast-to-cpu

 Disabled multicast addresses to cpu for PORT-VLAN 5 :
  224.0.0.5
  224.0.0.14 to 224.0.0.230

 Disabled multicast addresses to cpu for PORT-VLAN 10 :
  224.0.0.23

 Disabled multicast addresses to cpu for PORT-VLAN 20 :
  224.0.0.50 to 224.0.0.140
```

To display disabled multicast addresses for a particular VLAN, include the VLAN ID with the **show disabled-multicast-to-cpu** command. The following shows an example display.

```
FastIron# show disabled-multicast-to-cpu 5
 Disabled multicast addresses to cpu for PORT-VLAN 5 :

  224.0.0.5

  224.0.0.14 to 224.0.0.230
```

Syntax: show disabled-multicast-to-cpu [<vlan-id>]

For <vlan-id>, enter a valid VLAN ID. Note that each VLAN must have at least one port added to it.

Configuring a Static Multicast Route

Static multicast routes allow you to control the network path used by multicast traffic. Static multicast routes are especially useful when the unicast and multicast topologies of a network are different. You can avoid the need to make the topologies similar by instead configuring static multicast routes.

NOTE: This feature is not supported for DVMRP.

You can configure more than one static multicast route. The Layer 3 Switch always uses the most specific route that matches a multicast source address. Thus, if you want to configure a multicast static route for a specific multicast source and also configure another multicast static route for all other sources, you can configure two static routes as shown in the examples below.

To add static routes to multicast router A (see Figure 30.7), enter commands such as the following:

```
PIMRouterA(config)#ip mroute 1 207.95.10.0 255.255.255.0 interface ethernet 1/2
distance 1
PIMRouterA(config)#ip mroute 2 0.0.0.0 0.0.0.0 interface ethernet 2/3 distance 1
PIMRouterA(config)#write memory
```

Syntax: mroute <route-num> <ip-addr> interface ethernet [<slotnum>/]<portnum> | ve <num> [distance <num>]

or

Syntax: mroute <route-num> <ip-addr> rpf_address <rpf-num>

The <route-num> parameter specifies the route number.

The <ip-addr> command specifies the PIM source for the route.

NOTE: In IP multicasting, a route is handled in terms of its source, rather than its destination.

You can use the **ethernet** [<slotnum>/]<portnum> parameter to specify a physical port or the **ve** <num> parameter to specify a virtual interface.

NOTE: The **ethernet** [<slotnum>/]<portnum> parameter does not apply to PIM SM.

The **distance** <num> parameter sets the administrative distance for the route. When comparing multiple paths for a route, the Layer 3 Switch prefers the path with the lower administrative distance.

NOTE: Regardless of the administrative distances, the Layer 3 Switch always prefers directly connected routes over other routes.

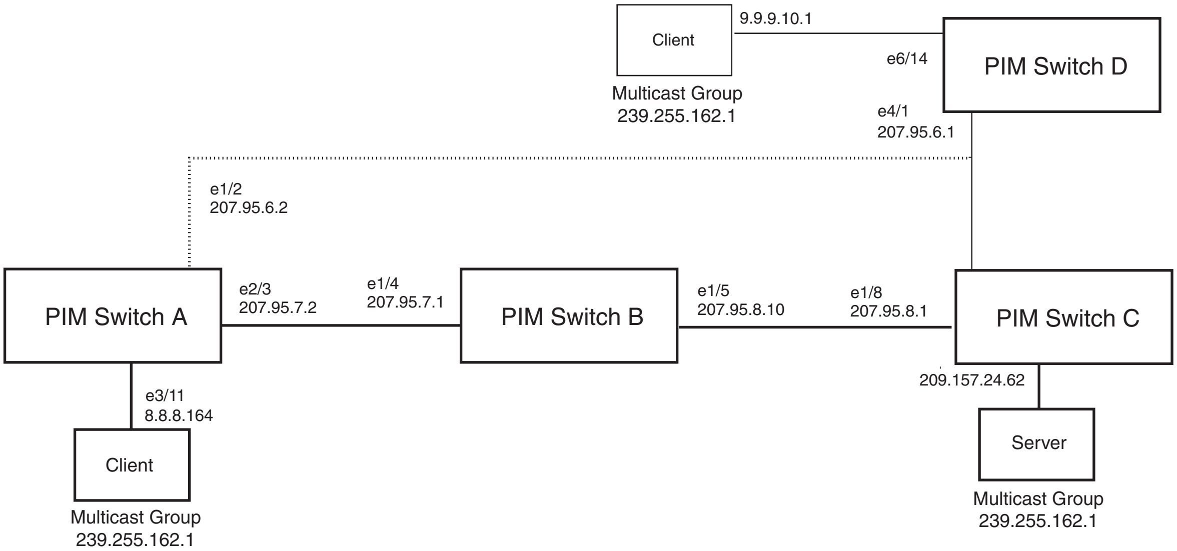
The **rpf_address** <rpf-num> parameter specifies an RPF number.

The example above configures two static multicast routes. The first route is for a specific source network, 207.95.10.0/24. If the Layer 3 Switch receives multicast traffic for network 207.95.10.0/24, the traffic must arrive on port 1/2. The second route is for all other multicast traffic. Traffic from multicast sources other than 207.95.10.0/24 must arrive on port 2/3.

Figure 30.7 shows an example of an IP Multicast network. The two static routes configured in the example above apply to this network. The commands in the example above configure PIM router A to accept PIM packets from 207.95.10.0/24 when they use the path that arrives at port 1/2, and accept all other PIM packets only when they use the path that arrives at port 2/3.

The distance parameter sets the administrative distance. This parameter is used by the software to determine the best path for the route. Thus, to ensure that the Layer 3 Switch uses the default static route, assign a low administrative distance value. When comparing multiple paths for a route, the Layer 3 Switch prefers the path with the lower administrative distance.

Figure 30.7 Example of Multicast Static Routes



To add a static route to a virtual interface, enter commands such as the following:

```
FastIron(config)#mroute 3 0.0.0.0 0.0.0.0 int ve 1 distance 1
FastIron(config)#write memory
```

Tracing a Multicast Route

The Foundry implementation of Mtrace is based on “A ‘traceroute’ facility for IP Multicast”, an Internet draft by S. Casner and B. Fenner. To trace a PIM route, use the following CLI method.

NOTE: This feature is not supported for DVMRP.

To trace a PIM route to PIM source 209.157.24.62 in group 239.255.162.1, enter a command such as the following:

```
FastIron#mtrace source 209.157.24.62 group 239.255.162.1

Type Control-c to abort
Tracing the route for tree 209.157.23.188

  0  207.95.7.2
  0  207.95.7.2 Thresh 0
  1  207.95.7.1 Thresh 0
  2  207.95.8.1 Thresh 0
  3  207.157.24.62
```

Syntax: mtrace source <ip-addr> group <multicast-group>

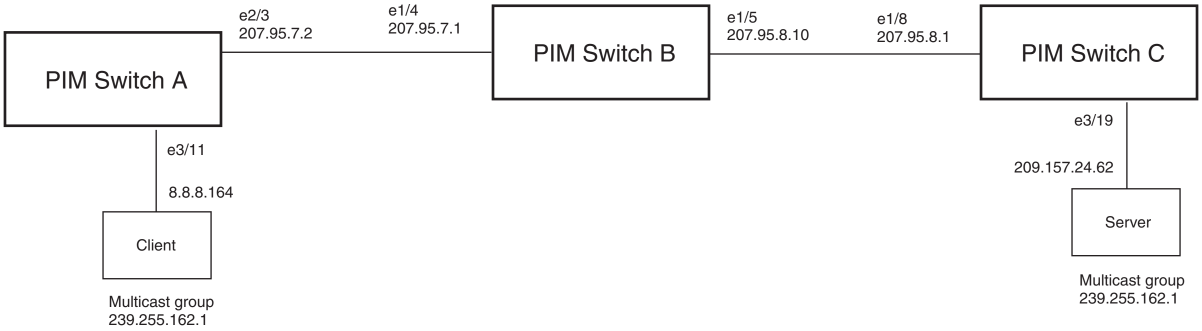
The **source** <ip-addr> parameter specifies the address of the route’s source.

NOTE: In IP multicasting, a route is handled in terms of its source, rather than its destination. When you trace an IP route, you specify its destination, but when you trace a PIM route, you specify its source.

The **group** <multicast-group> parameter specifies the PIM group the source IP address is in.

Figure 30.8 shows an example of an IP multicast group. The command example shown above is entered on PIM router A.

Figure 30.8 Example of a PIM Group



The command example above indicates that the source address 209.157.24.62 is three hops (three PIM switches) away from PIM Switch A. In PIM terms, each of the three switches has a forwarding state for the specified source address and multicast group. The value following “Thresh” in some of the lines indicates the TTL threshold. The threshold 0 means that all multicast packets are forwarded on the interface. If an administrator has set the TTL threshold to a higher value, only packets whose TTL is higher than the threshold are forwarded on the interface. The threshold is listed only for the PIM switch hops between the source and destination.

Displaying Another Multicast Router’s Multicast Configuration

The Foundry implementation of Mrinfo is based on the DVMRP Internet draft by T. Pusateri, but applies to PIM and not to DVMRP. To display the PIM configuration of another PIM router, use the following CLI method.

NOTE: This feature is not supported for DVMRP.

To display another PIM router’s PIM configuration, enter a command such as the following:

```
FastIron#mrinfo 207.95.8.1
207.95.8.1 -> 207.95.8.10 [PIM/0 /1]
207.95.10.2 -> 0.0.0.0 [PIM/0 /1 /leaf]
209.157.25.1 -> 0.0.0.0 [PIM/0 /1 /leaf]
209.157.24.1 -> 0.0.0.0 [PIM/0 /1 /leaf]
207.95.6.1 -> 0.0.0.0 [PIM/0 /1 /leaf]
128.2.0.1 -> 0.0.0.0 [PIM/0 /1 /leaf]
```

Syntax: mrinfo <ip-addr>

The <ip-addr> parameter specifies the IP address of the PIM router.

The output in this example is based on the PIM group shown in Figure 30.8 on page 30-46. The output shows the PIM interfaces configured on PIM router C (207.95.8.1). In this example, the PIM router has six PIM interfaces. One of the interfaces goes to PIM router B. The other interfaces go to leaf nodes, which are multicast end nodes attached to the router’s PIM interfaces. (For simplicity, the figure shows only one leaf node.)

When the arrow following an interface in the display points to a router address, this is the address of the next hop PIM router on that interface. In this example, PIM interface 207.95.8.1 on PIM router 207.95.8.1 is connected to PIM router 207.95.8.10. The connection can be a direct one or can take place through non-PIM routers. In this example, the PIM routers are directly connected.

When the arrow following an interface address points to zeros (0.0.0.0), the interface is not connected to a PIM router. The interface is instead connected to a leaf node.

NOTE: This display shows the PIM interface configuration information, but does not show the link states for the interfaces.

The information in brackets indicates the following:

- The multicast interface type (always PIM; this display is not supported for DVMRP)
- The Time-to-Live (TTL) for the interface.
- The metric for the interface
- Whether the interface is connected to a leaf node (“leaf” indicates a leaf node and blank indicates another PIM router)

For example, the information for the first interface listed in the display is “PIM/0 /1”. This information indicates that the interface is a PIM interface, has a TTL of 0, and a metric of 1. The interface is not a leaf node interface and thus is an interface to another PIM router.

The information for the second interface in the display is “PIM/0 /1/leaf”. This information indicates that the interface is a PIM interface, has a TTL of 0 and a metric of 1, and is connected to a leaf node.

IGMP V3

Platform Support:

- FESX/FSX/FWSX devices running software release 02.4.00 and later

The Internet Group Management Protocol (IGMP) allows an IPV4 interface to communicate IP Multicast group membership information to its neighboring routers. The routers in turn limit the multicast of IP packets with multicast destination addresses to only those interfaces on the router that are identified as IP Multicast group members. This release introduces the support of IGMP version 3 (IGMP V3) on Layer 3 Switches.

In IGMP V2, when a router sent a query to the interfaces, the clients on the interfaces respond with a membership report of multicast groups to the router. The router can then send traffic to these groups, regardless of the traffic source. When an interface no longer needs to receive traffic from a group, it sends a leave message to the router which in turn sends a group-specific query to that interface to see if any other clients on the same interface is still active.

In contrast, IGMP V3 provides selective filtering of traffic based on traffic source. A router running IGMP V3 sends queries to every multicast enabled interface at the specified interval. These queries determine if any interface wants to receive traffic from the router. The queries include the IP address of the traffic source (S) and/or the ID of the multicast group (G).

The interfaces respond to these queries by sending a membership report that contains one or more of the following records that are associated with a specific group:

- Current-State Record that indicates from which sources the interface wants to receive and not receive traffic. The record contains source address of interfaces and whether or not traffic will be received or included (IS_IN) or not received or excluded (IS_EX) from that source.
- Filter-mode-change record. If the interface changes its current state from IS_IN to IS_EX, a TO_EX record is included in the membership report. Likewise, if an interface’s current state changes from IS_EX to IS_IN, a TO_IN record appears in the membership report.

  IGMP V2 Leave report is equivalent to a TO_IN(empty) record in IGMP V3. This record means that no traffic from this group will be received regardless of the source.

  An IGMP V2 group report is equivalent to an IS_EX(empty) record in IGMP V3. This record means that all traffic from this group will be received regardless of source.
- Source-List-Change Record. If the interface wants to add or remove traffic sources from its membership report, the membership report can have an ALLOW record, which contains a list of new sources from which the interface wishes to receive traffic. It can also contains a BLOCK record, which lists current traffic sources from which the interfaces wants to stop receiving traffic.

In response to membership reports from the interfaces, the router sends a Group-Specific or a Group-and-Source Specific query to the multicast interfaces. Each query is sent three times with a one-second interval in between each transmission to ensure the interfaces receive the query. For example, a router receives a membership report with a Source-List-Change record to block old sources from an interface. The router sends Group-and-Source Specific Queries to the source and group (S,G) identified in the record. If none of the interfaces is interested in the (S,G), it is removed from (S,G) list for that interface on the router.

Each IGMP V3-enabled router maintains a record of the state of each group and each physical port within a virtual routing interface. This record contains the group, group-timer, filter mode, and source records information for the group or interface. Source records contain information on the source address of the packet and source timer. If the source timer expires when the state of the group or interface is in Include mode, the record is removed.

Default IGMP Version

IGMP V3 is available on devices running software release 02.4.00 and later; however, Foundry devices are shipped with IGMP V2 enabled. You must enable IGMP V3 globally or per interface.

Also, you must specify what version of IGMP you want to run on a device globally, on each interface (physical port or virtual routing interface), and on each physical port within a virtual routing interface. If you do not specify an IGMP version, IGMP V2 will be used.

Compatibility with IGMP V1 and V2

Different multicast groups, interfaces, and routers can run their own version of IGMP. Their version of IGMP is reflected in the membership reports that the interfaces send to the router. Routers and interfaces must be configured to recognized the version of IGMP you want them to process.

An interface or router sends the queries and reports that include its IGMP version specified on it. It may recognize a query or report that has a different version, but it may not process them. For example, an interface running IGMP V2 can recognize IGMP V3 packets, but cannot process them. Also, a router running IGMP V3 can recognize and process IGMP V2 packet, but when that router sends queries to an IGMP V2 interface, the host on that interface may not recognize the IGMP V3 queries. The interface or router does not automatically downgrade the IGMP version running on them to avoid version deadlock.

If an interface continuously receives queries from routers that are running versions of IGMP that are different from what is on the interface, the interface logs warning messages in the syslog every five minutes. Reports sent by interfaces to routers that contain different versions of IGMP do not trigger warning messages; however, you can see the versions of the packets using the **show ip igmp traffic** command.

The version of IGMP can be specified globally, per interface (physical port or virtual routing interface), and per physical port within a virtual routing interface. The IGMP version set on a physical port within a virtual routing interface supersedes the version set on a physical or virtual routing interface. Likewise, the version on a physical or virtual routing interface supersedes the version set globally on the device. The sections below present how to set the version of IGMP.

Globally Enabling the IGMP Version

Using the CLI

To globally identify the IGMP version on a Foundry device, enter the following command:

```
FastIron(config)#ip igmp version 3
```

Syntax: ip igmp version <version-number>

Enter 1, 2, or 3 for <version-number>. Version 2 is the default version.

Using the Web Management Interface

You cannot set the IGMP version using the Web management interface.

Enabling the IGMP Version Per Interface Setting

Using the CLI

To specify the IGMP version for a physical port, enter a command such as the following:

```
FastIron(config)#interface eth 1/5
FastIron(config-if-1/5)#ip igmp version 3
```

To specify the IGMP version for a virtual routing interface on a physical port, enter a command such as the following:

```
FastIron(config)#interface ve 3
FastIron(config-vif-1) ip igmp version 3
```

Syntax: [no] ip igmp version <version-number>

Enter 1, 2, or 3 for <version-number>. Version 2 is the default version.

Using the Web Management Interface

You cannot set the IGMP version using the Web management interface.

Enabling the IGMP Version on a Physical Port Within a Virtual Routing Interface

Using the CLI

To specify the IGMP version recognized by a physical port that is a member of a virtual routing interface, enter a command such as the following:

```
FastIron(config)#interface ve 3
FastIron(config-vif-3)#ip igmp version 2
FastIron(config-vif-3)#ip igmp port-version 3 e1/3-e1/7 e2/9
```

In this example, the second line sets IGMP V2 on virtual routing interface 3. However, the third line set IGMP V3 on ports 1/3 through 1/7 and port e2/9. All other ports in this virtual routing interface are configured with IGMP V2.

Syntax: ip igmp port-version <version-number> ethernet [<slotnum>/]<port-number>

Enter 1, 2, or 3 for <version-number>. IGMP V2 is the default version.

The **ethernet** <port-number> parameter specifies which physical port within a virtual routing interface is being configured. If you are entering this command on a chassis device, specify the slot number as well as the port number.

Using the Web Management Interface

You cannot set the IGMP version using the Web management interface.

Enabling Membership Tracking and Fast Leave

IGMP V3 provides membership tracking and fast leave to clients. In IGMP V2, only one client on an interface needs to respond to a router's queries; therefore, some of the clients may be invisible to the router, making it impossible for the router to track the membership of all clients in a group. Also, when a client leaves the group, the router sends group specific queries to the interface to see if other clients on that interface need the data stream of the client who is leaving. If no client responds, the router waits three seconds before it stops the traffic.

IGMP V3 contains the tracking and fast leave feature that you enable on virtual routing interfaces. Once enabled, all physical ports on that virtual routing interface will have the feature enabled. IGMP V3 requires all clients to respond to general and group specific queries so that all clients on an interface can be *tracked. Fast leave* allows clients to leave the group without the three second waiting period, if the following conditions are met:

- If the interface, to which the client belongs, has IGMP V3 clients only. Therefore, all physical ports on a virtual routing interface must have IGMP V3 enabled and no IGMP V1 or V2 clients can be on the interface. (Although IGMP V3 can handle V1 and V2 clients, these two clients cannot be on the interface in order for fast leave to take effect.)
- No other client on the interface is receiving traffic from the group to which the client belongs.

  Every group on the physical interface of a virtual routing interface keeps its own tracking record. However, it can track group membership only; it cannot track by (source, group).

For example, two clients (Client A and Client B) belong to group1 but each is receiving traffic streams from different sources. Client A receives a stream from (source_1, group1) and Client B receives it from (source_2, group1). The router still waits for three seconds before it stops the traffic because the two clients are in the same group. If the clients are in different groups, then the three second waiting period is not applied and traffic is stopped immediately. The **show ip igmp group tracking** command displays that clients in a group that are being tracked.

If a client sends a leave message, the client is immediately removed from the group. If a client does not send a report during the the specified group membership time (the default is 140 seconds), that client is removed from the tracking list.

USING THE CLI

To enable the tracking and fast leave feature, enter commands such as the following:

```
FastIron(config)#interface ve 13
FastIron(config-vif-13)#ip igmp tracking
```

Syntax: ip igmp tracking

USING THE WEB MANAGEMENT INTERFACE

You cannot change this parameter using the Web management interface.

Setting the Query Interval

The IGMP query interval period defines how often a router will query an interface for group membership. Possible values are 10 – 3,600 seconds and the default value is 60 seconds, but the value you enter must be a little more than twice the group membership time.

USING THE CLI

To modify the default value for the IGMP query interval, enter the following:

```
FastIron(config)#ip igmp query-interval 120
```

Syntax: ip igmp query-interval <10-3600>

The interval must be a little more than two times the group membership time.

USING THE WEB MANAGEMENT INTERFACE

If available, you can use the Web management interface to configure query interval. For example, log in to the Web management interface and go to the Configure -> DVMRP -> IGMP panel. Enter a value from 10 – 3600 in the Query Interval field

Setting the Group Membership Time

Group membership time defines how long a group will remain active on an interface in the absence of a group report. Possible values are from 20 – 7200 seconds and the default value is 140 seconds.

USING THE CLI

To define an IGMP membership time of 240 seconds, enter the following:

```
FastIron(config)#ip igmp group-membership-time 240
```

Syntax: ip igmp group-membership-time <20-7200>

USING THE WEB MANAGEMENT INTERFACE

If available, you can use the Web management interface to configure group membership time. For example, log in to the Web management interface and go to the Configure -> DVMRP -> IGMP panel. Enter a value from 20 – 7200 in the Group Membership Time field.

Setting the Maximum Response Time

The maximum response time defines the maximum number of seconds that a client can wait before it replies to the query sent by the router. Possible values are 1 – 10. The default is 5.

USING THE CLI

To change the IGMP maximum response time, enter a command such as the following at the global CONFIG level of the CLI:

```
FastIron(config)#ip igmp max-response-time 8
```

Syntax: [no] ip igmp max-response-time <num>

The <num> parameter specifies the maximum number of seconds for the response time. Enter a value from 1 – 10. The default is 5.

USING THE WEB MANAGEMENT INTERFACE

You cannot change this parameter using the Web management interface.

IGMP V3 and Source Specific Multicast Protocols

Enabling IGMP V3 enables source specific multicast (SSM) filtering for DVMRP and PIM Dense (PIM-DM) for multicast group addresses in the 224.0.1.0 through 239.255.255.255 address range. However, if PIM Sparse is used as the multicast protocol, the SSM protocol should be enabled if you want to filter unwanted traffic before the Shortest Path Tree protocol switchover occurs for groups in the 232/8 range. Not configuring the SSM protocol in PIM Sparse may cause the switch or router to leak unwanted packets with the same group, but containing undesired sources, to clients. After SPT switch over, the leak stops and source specific multicast works correctly even without configuring the SSM protocol.

If the SSM protocol is not enabled and before the SPT switchover, the multicast router creates one (*, G) entry for the entire multicast group, which can have many sources. If the SSM protocol is enabled, one (S,G) entry is created for every member of the multicast group, even for members with non-existent traffic. For example, if there are 1,000 members in the group, 1,000 (S,G) entries will be created. Therefore, enabling the SSM protocol for PIM-SM requires more resources than leaving the protocol disabled.

Enabling SSM

To enable the SSM protocol on a Foundry device running PIM-SM, enter a command such as the following:

```
FastIron(config)#router pim
FastIron(config-pim-router)#ssm-enable
```

Syntax: [no] ssm-enable

Enter the ssm-enable command under the router pim level to globally enable the SSM protocol on a Layer 3 Switch.

Displaying IGMP V3 Information on Layer 3 Switches

The sections below present the show commands available for IGMP V3 on Layer 3 Switches. For show commands on Layer 2 Switches, use the **show ip multicast** commands which are discussed in the section “IGMP Snooping Show Commands” on page 27-15.

Displaying IGMP Group Status

NOTE: This report is available on Layer 3 Switches.

To display the status of all IGMP multicast groups on a device, enter the following command:

```
FastIron#show ip igmp group
Interface v18 : 1 groups
      group            phy-port static querier life mode    #_src
1     239.0.0.1        e4/20    no     yes          include 19
Interface v110 : 3 groups
      group            phy-port static querier life mode    #_src
2     239.0.0.1        e4/5     no     yes          include 10
3     239.0.0.1        e4/6     no     yes     100  exclude 13
4     224.1.10.1       e4/5     no     yes          include 1
```

To display the status of one IGMP multicast group, enter a command such as the following:

```
FastIron#show ip igmp group 239.0.0.1 detail
Display group 239.0.0.1 in all interfaces.
Interface v18 : 1 groups
      group           phy-port static querier life mode    #_src
1     239.0.0.1        e4/20    no     yes          include 19
    group: 239.0.0.1, include, permit 19 (source, life):
      (3.3.3.1 40) (3.3.3.2 40) (3.3.3.3 40) (3.3.3.4 40) (3.3.3.5 40)
      (3.3.3.6 40) (3.3.3.7 40) (3.3.3.8 40) (3.3.3.9 40) (3.3.3.10 40)
      (3.3.3.11 40) (3.3.3.12 40) (3.3.3.13 40) (3.3.3.14 40) (3.3.3.15 40)
      (3.3.3.16 40) (3.3.3.17 40) (3.3.3.18 40) (3.3.3.19 40)
Interface v110 : 1 groups
      group           phy-port static querier life mode    #_src
2     239.0.0.1        e4/5     no     yes          include 10
    group: 239.0.0.1, include, permit 10 (source, life):
      (2.2.3.0 80) (2.2.3.1 80) (2.2.3.2 80) (2.2.3.3 80) (2.2.3.4 80)
      (2.2.3.5 80) (2.2.3.6 80) (2.2.3.7 80) (2.2.3.8 80) (2.2.3.9 80)
```

If the tracking and fast leave feature is enabled, you can display the list of clients that belong to a particular group by entering commands such as the following:

```
FastIron#show ip igmp group 224.1.10.1 tracking
Display group 224.1.10.1 in all interfaces with tracking enabled.
Interface v13 : 1 groups, tracking_enabled
      group           phy-port static querier life mode    #_src
1     224.1.10.1       e4/15    no     yes          include 3
    receive reports from 3 clients:
      110.110.110.7 110.110.110.8 110.110.110.9
```

Syntax: show ip igmp group [<group-address> [detail] [tracking]]

If you want a report for a specific multicast group, enter that group's address for <group-address>. Omit the <group-address> if you want a report for all multicast groups.

Enter **detail** if you want to display the source list of the multicast group.

Enter **tracking** if you want information on interfaces that have tracking enabled.

The following table defines the statistics for the **show ip igmp group** command output.

Table 30.14: Output of show ip igmp group

This Field	Displays
Group	The address of the multicast group
Phy-port	The physical port on which the multicast group was received.
Static	A "yes" entry in this column indicates that the multicast group was configured as a static group; "No" means it was not. Static multicast groups can be configured in IGMP V2 using the **ip igmp static** command. In IGMP V3, static sources cannot be configured in static groups.

Table 30.14: Output of show ip igmp group

This Field	Displays
Querier	“Yes” means that the port is a querier port; “No” means it is not. A port becomes a non-querier port when it receives a query from a source with a lower source IP address than the port.
Life	Shows the number of seconds the interface can remain in exclude mode. An exclude mode changes to include mode if it does not receive an "IS_EX" or "TO_EX" message during a certain period of time. The default is 140 seconds. There is no "life" displayed in include mode.
Mode	Indicates current mode of the interface: Include or Exclude. If the interface is in Include mode, it admits traffic only from the source list. If an interface is in Exclude mode, it denies traffic from the source list and accepts the rest.
#_src	Identifies the source list that will be included or excluded on the interface. If IGMP V2 group is in Exclude mode with a #_src of 0, the group excludes traffic from 0 (zero) source list, which means that all traffic sources are included.
Group:	If you requested a *detailed* report, the following information is displayed: • The multicast group address • The mode of the group • A list of sources from which traffic will be admitted (include) or denied (exclude) on the interface is listed. • The life of each source list. If you requested a *tracking* report, the clients from which reports were received are identified.

Displaying the IGMP Status of an Interface

You can display the status of a multicast enabled port by entering a command such as the following:

NOTE: This report is available on Layer 3 Switches.

```
FastIron#show ip igmp interface
query interval = 60, max response time= 3, group membership time=140
v5: default V2,         PIM dense, addr=1.1.1.2
  e4/12   has    0 groups, non-Querier (age=40), default V2
v18: default V2,        DVMRP, addr=2.2.2.1
  e4/20   has    0 groups, Querier, default V2
v20: configured V3,     PIM dense (port down), addr=1.1.20.1
v110: configured V3,    PIM dense, addr=110.110.110.1
  e4/6    has    2 groups, Querier, default V3
    group: 239.0.0.1, exclude, life=100, deny 13
    group: 224.1.10.1, include, permit 2
  e4/5    has    3 groups, Querier, default V3
    group: 224.2.2.2, include, permit 100
    group: 239.0.0.1, include, permit 10
    group: 224.1.10.1, include, permit 1
```

Syntax: show ip igmp interface [ve | ethernet <number> <group-address>]

Enter **ve** and its <number> or **ethernet** and its <number> to display information for a specific virtual routing interface or ethernet interface.

Entering an address for <group-address> displays information for a specified group on the specified interface.

The report shows the following information:

Table 30.15: Output of show ip igmp interface

This Field	Displays
Query interval	Displays how often a querier sends a general query on the interface.
Max response	The maximum number of seconds a client can wait before it replies to the query.
Group membership time	The number of seconds multicast groups can be members of this group before aging out.
(details)	The following is displayed for each interface: • The ID of the interface • The IGMP version that it is running (default IGMP V2 or configured IGMP V3) • The multicast protocol it is running: DVMRP, PIM-DM, PIM-SM • Address of the multicast group on the interface • If the interface is a virtual routing interface, the physical port to which that interface belongs, the number of groups on that physical port, whether or not the port is a querier or a non-querier port, the age of the port, and other multicast information for the port are displayed.

Displaying IGMP Traffic Status

To display the traffic status on each virtual routing interface, enter the following command:

NOTE: This report is available on Layer 3 Switches.

```
FastIron#show ip igmp traffic
Recv   QryV2 QryV3 G-Qry GSQry MbrV2 MbrV3 Leave  IsIN   IsEX ToIN ToEX ALLOW BLK
v5        29     0     0     0     0     0     0     0     0    0    0     0    0
v18       15     0     0     0     0    30     0    60     0    0    0     0    0
v110       0     0     0     0     0    97     0   142    37    2    2     3    2
Send   QryV1 QryV2 QryV3 G-Qry GSQry
v5         0     2     0     0     0
v18        0     0    30    30     0
v110       0     0    30    44    11
```

Syntax: show ip igmp traffic

The report shows the following information:

Table 30.16: Output of show ip igmp traffic

This Field	Displays
QryV2	Number of general IGMP V2 query received or sent by the virtual routing interface.
QryV3	Number of general IGMP V3 query received or sent by the virtual routing interface.
G-Qry	Number of group specific query received or sent by the virtual routing interface.
GSQry	Number of source specific query received or sent by the virtual routing interface.
MbrV2	The IGMP V2 membership report.
MbrV3	The IGMP V3 membership report.
Leave	Number of IGMP V2 “leave” messages on the interface. (See ToEx for IGMP V3.)
IsIN	Number of source addresses that were included in the traffic.
IsEX	Number of source addresses that were excluded in the traffic.
ToIN	Number of times the interface mode changed from exclude to include.
ToEX	Number of times the interface mode changed from include to exclude.
ALLOW	Number of times that additional source addresses were allowed or denied on the interface:
BLK	Number of times that sources were removed from an interface.

Clearing IGMP Statistics

To clear statistics for IGMP traffic, enter the following command:

```
FastIron#clear igmp traffic
```

Syntax: clear igmp traffic

This command clears all the multicast traffic information on all interfaces on the device.

IGMP Proxy

Platform Support: FastIron X Series devices running software release 03.2.00 or later

IGMP Proxy provides a means for the FastIron X Series routers to receive any or all multicast traffic from an upstream device if the router is not able to run PIM.

IGMP Proxy enables the router to issue IGMP host messages on behalf of hosts that the router discovered through standard PIM interfaces. The router acts as a proxy for its hosts and performs the host portion of the IGMP task on the upstream interface as follows:

- When queried, the router sends group membership reports for the groups learned
- When one of its hosts joins a multicast address group to which none of its other hosts belong, the router sends unsolicited membership reports to that group.
- When the last of its hosts in a particular multicast group leaves the group, the FastIron X Series router sends an unsolicited leave group membership report to group for all routers (multicast IP address 244.0.0.2)

Configuration Notes

When using IGMP Proxy, you must:

1. Configure PIM on all multicast client ports to build the group membership table. The group membership table will be reported by the proxy interface. See “Globally Enabling and Disabling PIM” on page 30-8.
2. Enable IP multicast on an interface to an upstream FastIron X Series router that will be the IGMP proxy interface and configure IGMP Proxy on that interface

Also note the following limitations:

- IGMP Proxy is available in full-Layer 3 images, starting with FSX software release 03.2.00.
- IGMP Proxy cannot be enabled on the same interface on which PIM SM, PIM DM, or DVMRP is enabled.
- IGMP Proxy is only supported in a PIM Dense environment where there are IGMP clients connected to the Foundry device. The Foundry device will not send IGMP reports on an IGMP proxy interface for remote clients connected to a PIM neighbor, as it will not be aware of groups that the remote clients are interested in.

Configuring IGMP Proxy

To configure IGMP Proxy:

1. Configure router PIM globally:

   ```
   FastIron(config)#router pim
   ```

2. Configure an IP address on the interface (physical or virtual routing interface) that will serve as the IGMP proxy for an upstream device by entering commands such as the following:

   ```
   FastIron(config)#int e 1/3
   FastIron(config-if-e1000-1/3)#ip address 207.95.5.1/24
   ```

3. Enable IGMP Proxy on the interface.

   ```
   FastIron(config-if-e1000-1/3)#ip igmp proxy
   ```

Syntax: [no] ip igmp proxy

Once IGMP Proxy is configured and the FastIron X Series router receives a query on an IGMP Proxy interface, the router sends a report in response to the query before the IGMP maximum response time expires.

Displaying IGMP Proxy Traffic

Use the **show ip igmp traffic** command to see traffic for IGMP Proxy.

```
FastIron#show ip igmp traffic

Recv  QryV2 QryV3 G-Qry GSQry MbrV2 MbrV3 Leave  IsIN  IsEX ToIN ToEX ALLO  BLK
e1/14     0     0     0     0 27251     0    12      0 27251    12    0    0    0
v10     250     0     0     0   244     0     0      0   244     0    0    0    0
Send  QryV1 QryV2 QryV3 G-Qry GSQry MbrV1 Mbrv2 Leave
e1/14     0  1365     0    48     0     0     0     0
v10       0     1     0     0     0     0 25602     1
```

Syntax: show ip igmp traffic

See “Displaying IGMP Traffic Status” on page 30-55 to interpret the information in the output. The fields in bold show information for IGMP Proxy.

IP Multicast Protocols and IGMP Snooping on the Same Device

Platform Support: FastIron X Series devices running software release 04.1.00 and later – L2, BL3, L3

Software release FSX 04.1.00 adds support for global Layer 2 IP multicast traffic reduction (IGMP snooping) and Layer 3 multicast routing (DVMRP/PIM-Sparse/PIM-Dense) together on the same device in the full Layer 3 software image, as long as the Layer 2 feature configuration is at the VLAN level. Releases prior to FSX 04.1.00 support global Layer 2 IP multicast traffic reduction and Layer 3 multicast routing. However, they were mutually exclusive and configuring IGMP snooping on a VLAN together with DVMRP/PIM-Sparse/PIM-Dense was not allowed.

For Layer 2 multicast traffic reduction, IGMP snooping is performed independently within all VLANs that have the feature configured. Layer 3 multicast routing is performed between the IP interfaces that are configured for DVMRP/PIM-Sparse/PIM-Dense. A Layer 3 interface could be a physical, loopback, or VE port configured with an IP address.

If there are two sources for a single group, where one source sends traffic into a VLAN with IGMP snooping enabled, while the other source sends traffic to a PIM enabled Layer 3 interface, a client for the group in the same VLAN as the first source will only receive traffic from that source. It will not receive traffic from the second source connected to the Layer 3 interface. Similarly, if there is another IP interface with a Layer 3 client or PIM/DVMRP neighbor that requests traffic for the same group, it will only receive traffic from the second source and not the first.

Configuration Example

Figure 30.9 and Figure 30.10 show an example IGMP snooping and PIM forwarding configuration.

Figure 30.9 Example 1: IGMP Snooping and PIM Forwarding

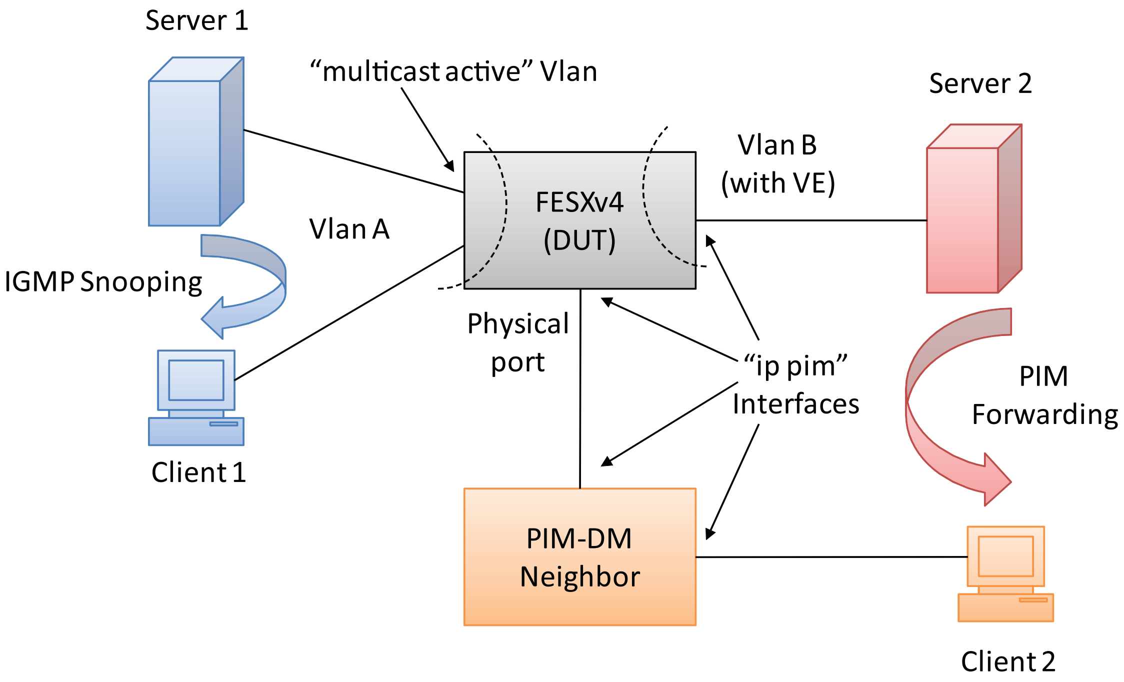
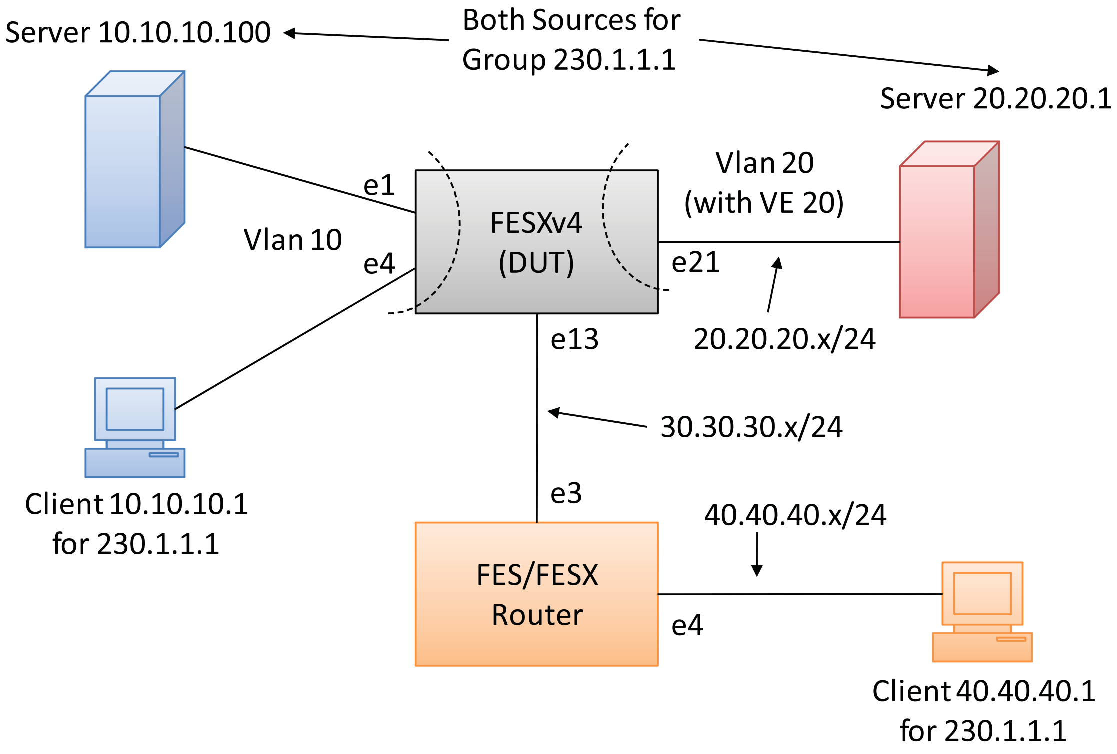


Figure 30.10 Example 2: IGMP Snooping and PIM Forwarding



CLI Commands

The following are the CLI commands for the configuration example shown in Figure 30.9 and Figure 30.10.

1. On the FESXv4 device, configure IGMP Snooping on VLAN 10:

```
FastIron(config)#vlan 10 by port
FastIron(config-vlan-10)#untagged e 1 to 4
Added untagged port(s) ethe 1 to 4 to port-vlan 10.
FastIron(config-vlan-10)#router-interface ve 10
FastIron(config-vlan-10)#ip multicast active
FastIron(config-vlan-10)#interface ve 10
FastIron(config-vif-10)#ip address 10.10.10.10/24
```

2. On the FESXv4 device, enable PIM routing between VLAN/VE 20 and Interface e 13:

```
FastIron(config)#vlan 20 by port
FastIron(config-vlan-20)#untagged e 21 to 24
Added untagged port(s) ethe 21 to 24 to port-vlan 20.
FastIron(config-vlan-20)#router-interface ve 20
FastIron(config-vlan-20)#exit
FastIron(config)#router pim
FastIron(config-pim-router)#exit
FastIron(config)#interface ve 20
FastIron(config-vif-20)#ip address 20.20.20.10/24
FastIron(config-vif-20)#ip pim
FastIron(config-vif-20)#exit
FastIron(config)#interface e 13
FastIron(config-if-e1000-13)#ip address 30.30.30.10/24
FastIron(config-if-e1000-13)#ip pim
```

3. Configure the FES/FESX neighboring device:

```
FastIron(config)#ip route 20.20.20.0 255.255.255.0 30.30.30.10
FastIron(config)#router pim
FastIron(config-pim-router)#exit
FastIron(config)#interface ethernet 3
FastIron(config-if-e1000-3)#ip address 30.30.30.20/24
FastIron(config-if-e1000-3)#ip pim
FastIron(config-if-e1000-3)#interface ethernet 4
FastIron(config-if-e1000-4)#ip address 40.40.40.20/24
FastIron(config-if-e1000-4)#ip pim
```

Chapter 31

Configuring Multicast Listening Discovery (MLD) Snooping on the FGS and FLS

This chapter describes how to configure Multicast Listening Discovery (MLD) Snooping on Foundry FGS and FLS devices running software release 03.0.00 and later (on devices running IPv6).

Overview

The default method a FastIron uses to process an IPv6 multicast packet is to broadcast it to all ports except the incoming port of a VLAN. Packets are flooded by hardware without going to CPU, which may result in some clients receiving unwanted traffic.

MLD Snooping provides multicast containment by forwarding traffic only to those clients that have MLD receivers for a specific multicast group (destination address). The FastIron maintains the MLD group membership information by processing MLD reports and generating messages so traffic can be forwarded to ports receiving MLD reports. This is analogous to IGMP Snooping on the Foundry Layer3 switches.

An IPv6 multicast address is a destination address in the range of FF00::/8. A limited number of multicast addresses are reserved. Since packets destined for the reserved addresses may require VLAN flooding, FGS and FLS devices don't snoop in the FF0X::000X range (where X is from 0 to F). Data packets destined to these addresses are flooded to the entire VLAN by hardware, and mirrored to CPU. Multicast data packets destined to addresses outside the FF0X::000X range are snooped. A client must send MLD reports in order to receive traffic. If an application outside the FF0X::000X range requires VLAN flooding, you must configure a static group for the entire VLAN.

An MLD device periodically broadcasts general queries, and sends group queries upon receiving a leave message to ensure no other clients at the same port still want this specific traffic before removing it. MLDv1 allows clients to specify which group (destination IPv6 address) on which to receive traffic. (MLDv1 cannot choose the source of the traffic.) MLDv2 deals with source-specific multicasts, adding the capability for clients to INCLUDE or EXCLUDE specific traffic sources. An MLDv2 device's port state can either be in INCLUDE or EXCLUDE mode. There are different types of group records for client reports.

The interfaces respond to general queries by sending a membership report containing one or more of the following records associated with a specific group:

- Current-state record - Indicates the sources from which the interface wants to receive or not receive traffic. This record contains the source addresses of the interfaces and whether or not traffic will be included (IS_IN) or excluded (IS_EX) from that source address.
- Filter-mode-change record - If the interface changes its current state from IS_IN to IS_EX, a TO_EX record is included in the membership report. Likewise, if an interface's current state changes from IS_EX to IS_IN, a TO_IN record appears in the membership report.
- MLDv1 leave report - Equivalent to a TO_IN (empty) record in MLDv2. This record means that no traffic from this group will be received regardless of the source.

- An MLDv1 group report - Equivalent to an IS_EX (empty) record in MLDv2. This record means that all traffic from this group will be received regardless of source.
- Source-list-change record - If the interface wants to add or remove traffic sources from its membership report, the report can include an ALLOW record, which contains a list of new sources from which the interface wishes to receive traffic. The report can also contain a BLOCK record, which lists current traffic sources from which the interface wants to stop receiving traffic.

MLD protocols provide a way for clients and a device to exchange messages, and allow the device to build a database indicating which port wants what traffic. Since the MLD protocols do not specify forwarding methods, MLD Snooping or multicast protocols such as IPv6 PIM-Sparse Mode (PIM SM) are required to handle packet forwarding. PIM SM can route multicast packets within and outside a VLAN, while MLD Snooping can switch packets only within a VLAN. FGS and FLS devices do not support PIM-SM routing.

If a VLAN is not MLD Snooping-enabled, it floods IPv6 multicast data and control packets to the entire VLAN in hardware. When snooping is enabled, MLD packets are trapped to the CPU. Data packets are mirrored to the CPU and VLAN flooded. The CPU then installs hardware resources so subsequent data packets can be hardware-switched to desired ports without going through the CPU. If there is no client report, the hardware resource drops the data stream. The hardware can either match group addresses only (* G), or both source and group (S G) addresses in the data stream. If MLDv2 is configured in any port of a VLAN, the VLAN uses an (S G) match, otherwise it uses (* G). Because the FGS and FLS hardware can match only the lowest 32 bits of a 128 bit IPv6 address, the output interfaces (OIF) of a hardware resource are the superset of the OIF of all data streams sharing the same lowest 32 bits. For example, if groups ff10::1234:5678:abcd and ff20::5678:abcd share the same hardware resource, then the OIF of the hardware matching (* 5678:abcd) is the superset of these two groups.

An FGS or FLS device allocates 16K of hardware resources for MAC learning, IGMP, and MLD snooping. If a data packet does not match any of these resources, it might be sent to the CPU, increasing the CPU burden. This can happen if the device runs out of hardware resources, or is unable to install a resource for a specific matching address due to a hashing collision. Because the hardware hashes addresses into 16K entries, some addresses may be hashed into the same entry. If the collision number in an entry is more than the hardware chain length, the resource cannot be installed. The chain length can be configured using the **hash-chain-length** command, as follows:

```
FastIron(config)#hash-chain-length 8
```

Syntax: [no] hash-chain-length <num>

The <num> parameter range is 4 to 32, in multiples of 4. If the input value is not a multiple of 4, then it will be changed to the multiple of 4 lower than then the input value (e.g. 11 will be changed to 8). The default hash chain length is 4. A chain length of more than 4 may affect line rate switching.

NOTE: For this command to take effect, you must save the configuration and reload the switch.

The hardware resource limit applies only to snooping-enabled VLANs. In VLANs where snooping is not enabled, multicast streams are switched in hardware without using any pre-installed resources.

An FGS or FLS device supports up to 32K of MLD groups. They are produced by client membership reports.

Configuration Notes:

- Servers (traffic sources) are not required to send MLD memberships.
- The default MLD version is V1.
- Hardware resources are installed only when there is data traffic. If a VLAN is configured for MLDv2, the hardware matches (S G), otherwise it matches (* G).
- You can configure the maximum number of groups and hardware-switched data streams.
- The device supports static groups applying to the entire VLAN, or to specific ports. The device acts as a proxy to send MLD reports for the static groups when receiving queries.
- A user can configure static router ports, forcing all multicast traffic to be sent to these ports.

- Foundry FastIron GS and FastIron LS devices support fast leave for MLDv1, which stops traffic immediately to any port that has received a leave message.
- Foundry FastIron GS and FastIron LS devices support tracking and fast leave for MLDv2, which tracks all MLDv2 clients. If the only client on a port leaves, traffic is stopped immediately.
- An MLD device can be configured as a querier (active) or non-querier (passive). Queriers send queries. Non-queriers listen for queries and forward them to the entire VLAN.
- Every VLAN can be independently configured as a querier or a non-querier.
- A VLAN that has a connection to an IPv6 PIM-enabled port on another router should be configured as a non-querier. When multiple snooping devices connect together and there is no connection to IPv6 PIM ports, only one device should be configured as the querier. If multiple devices are configured as active, only one will continue to send queries after the devices have exchanged queries. See “Configuring Queriers and Non-Queriers” on page 31-3.
- An MLD device can be configured to rate-limit the forwarding of MLDv1 membership reports to queriers.
- Because FGS and FLS devices use an IPv6 link-local address as the source address when sending queries, no global address is required.

The MLD implementation allows snooping on some VLANs or on all VLANs. MLD can be enabled or disabled independently for each VLAN. In addition, individual ports of a VLAN can be configured as MSLv1 and MLDv2. In general, global configuration commands such as **ipv6 mld-snooping..** apply to all VLANs except those with a local **mld-snooping..** configuration, which supersedes the global configuration. Configuring the version on a port or a VLAN only affects the device’s sent query version. The device always processes all versions of client reports regardless of the version configured.

MLD Snooping requires hardware resources. If the device has insufficient resources, the data stream without a resource is mirrored to the CPU in addition to being VLAN flooded, which can cause high CPU usage. To avoid this situation, Foundry recommends that you avoid enabling snooping globally unless necessary.

When any port of a VLAN is configured for MLDv2, the VLAN matches both source and group (S G) in hardware switching. If no port is configured for MLDv2, the VLAN matches group only (* G). Matching (S G) requires more hardware resources than (* G) when there are multiple servers sharing the same group. For example, two data streams from different sources to the same group require two (S G) entries in MLDv2, compared to only one (* G) in MLDv1. Foundry recommends that you use MLDv2 only in a source-specific application. Because each VLAN can be configured for the version independently, some VLANs might match (* G) while others match (S G).

To receive data traffic, MLD Snooping requires clients to send membership reports. If a client does not send reports, you must configure a static group to force traffic to client ports. The static group can either apply to some ports or to the entire VLAN.

Configuring Queriers and Non-Queriers

An MLD Snooping-enabled FGS device can be configured as a querier (active) or non-querier (passive). An MLD querier sends queries; a non-querier listens for MLD queries and forwards them to the entire VLAN. Starting with release 03.0.00, VLANs can be independently configured as queriers or non-queriers. If a VLAN has a connection to an IPv6 PIM-enabled port on another router, the VLAN should be configured as a non-querier. When multiple MLD snooping devices are connected together, and there is no connection to an IPv6 PIM-enabled port, one of the devices should be configured as a querier. If multiple devices are configured as queriers, after multiple devices exchange queries, then all devices except the winner (the device with the lowest address) stop sending queries. Although the system works when multiple devices are configured as queriers, Foundry recommends that only one device, preferably the one with the traffic source, is configured as the querier.

Because non-queriers always forward multicast data traffic and MLD messages to router ports which receive MLD queries or IPv6 PIM hellos, Foundry recommends that you configure the devices with the data traffic source (server) as queriers. If a server is attached to a non-querier, the non-querier always forwards traffic to the querier regardless of whether or not there are clients on the querier.

NOTE: In a topology with one or more connected devices, at least one device must be running PIM, or configured as active. Otherwise, no devices can send queries, and traffic cannot be forwarded to clients.

VLAN Specific Configuration

You can configure MLD snooping on some VLANs or all VLANs. Each VLAN can be independently enabled or disabled for MLD snooping, or can be configured with MLDv1 or MLDv2. In general, the **ipv6 mld-snooping...** commands apply globally to all VLANs except those configured with VLAN-specific **mld-snooping...** commands. VLAN-specific **mld-snooping** commands supersede global **ipv6 mld-snooping** commands.

Using MLDv1 with MLDv2

MLD snooping can be configured as MLDv1 or MLDv2 on individual ports on a VLAN. An interface or router sends queries and reports that include the MLD version with which it has been configured. The version configuration applies only to the sending of queries. The snooping device recognizes and processes MLDv1 and MLDv2 packets regardless of the version configured.

NOTE: To avoid version deadlock, when an interface receives a report with a lower version than that for which it has been configured, the interface does **not** automatically downgrade the running MLD version.

Configuring MLD Snooping

Configuring MLD Snooping on an IPv6 FGS device consists of the following global and VLAN-specific tasks:

Global Tasks:

- Configuring hardware and software resource limits
- Disabling transmission and receipt of MLD packets on a port
- Configuring the MLD mode: active or passive (must be enabled for MLD Snooping)
- Modifying the age interval
- Specifying the interval for query messages (active MLD mode only)
- Specifying the global MLD version
- Enabling and disabling report control (rate limiting)
- Modifying the leave-wait time
- Modifying the mcache age interval
- Disabling error and warning messages

VLAN-Specific Tasks:

- Configuring the MLD mode for the VLAN: active or passive
- Enabling or disabling MLD Snooping for the VLAN
- Configuring the MLD version for the VLAN
- Configuring the MLD version for individual ports in the VLAN
- Configuring static groups to the entire VLAN or some ports
- Configuring static router ports
- Enabling client tracking and the fast leave feature for MLDv2
- Configuring fast leave for MLDv1
- Configuring fast-convergence

Configuring the Hardware and Software Resource Limits

The system supports up to 8K of hardware-switched multicast streams. The configurable range is from 256 to 8192 and the default is 4096. Enter a command such as the following to define the maximum number of MLD Snooping cache entries:

```
FastIron(config)#system-max mld-snoop-mcache 8000
```

Syntax: [no] system-max mld-snoop-mcache <num>

The system supports up to 32K of groups. The configurable range is 256 to 32768 and the default is 8192. The configured number is the upper limit of an expandable database. Client memberships exceeding the group limits are not processed.

Disabling Transmission and Receipt of MLD packets on a Port

When a VLAN is snooping-enabled, all MLD packets are trapped to the CPU without hardware VLAN flooding. The CPU can block MLD packets to and from a multicast-disabled port, and will not add that port to the output interfaces of hardware resources, which prevents the disabled port from receiving multicast traffic. However, if static groups to the entire VLAN are defined, the traffic for these groups is flooded to the entire VLAN, including to the disabled ports. Since the hardware cannot block traffic from disabled ports, hardware traffic is switched in the same way as traffic from enabled ports.

NOTE: This command has no effect on a VLAN that is not snooping-enabled because all multicast traffic is VLAN flooded.

```
FastIron(config)#interface ethernet 0/1/3
FastIron(config-if-e1000-0/1/3)#ipv6-multicast-disable
```

Syntax: [no] ipv6-multicast-disable

Configuring the Global MLD Mode

You can configure an FGS or FLS device for either active or passive (default) MLD mode. If you specify an MLD mode for a VLAN, the MLD mode overrides the global setting.

- Active – In active MLD mode, an FGS device actively sends out MLD queries to identify IPv6 multicast groups on the network, and makes entries in the MLD table based on the group membership reports it receives from the network.
- Passive – In passive MLD mode, the device forwards reports to the router ports which receive queries. MLD Snooping in passive mode does not send queries, but does forward queries to the entire VLAN.

To globally set the MLD mode to active for the FGS or FLS device, enter the following command:

```
FastIron(config)#ipv6 mld-snooping active
```

Syntax: [no] ipv6 mld-snooping [active | passive]

Omitting both the **active** and **passive** keywords is the same as entering **ipv6 mld-snooping passive**.

Modifying the Age Interval

When the FGS device receives a group membership report, it makes an entry in the MLD group table for the group in the report. The age interval specifies how long the entry can remain in the table without the device receiving another group membership report. When multiple devices connect together, all devices should be configured with the same age interval. The age interval should be at least twice that of the query interval, so that missing one report won't stop traffic. For a non-querier, the query interval should equal that of the querier.

To modify the age interval, enter a command such as the following:

```
FastIron(config)#ipv6 mld-snooping age-interval 280
```

Syntax: [no] ipv6 mld-snooping age-interval <interval>

The <interval> parameter specifies the aging time. You can specify a value from 20 – 7200 seconds. The default is 140 seconds.

Modifying the Query Interval (Active MLD Snooping Mode Only)

If the MLD mode is set to active, you can modify the query interval, which specifies how often an FGS or FLS device sends group membership queries. When multiple queriers connect together, all queriers should be configured with the same interval.

To modify the query interval, enter a command such as the following:

```
FastIron(config)#ipv6 mld-snooping query-interval 120
```

Syntax: [no] ipv6 mld-snooping query-interval <interval>

The <interval> parameter specifies the interval between queries. You can specify a value from 10 – 3600 seconds. The default is 60 seconds.

Configuring the Global MLD Version

The default version is MLDv1. You can specify the global MLD version on the FGS or FLS device as either MLDv1 or MLDv2. For example, the following command configures the device to use MLDv2:

```
FastIron(config)#ipv6 mld-snooping version 2
```

Syntax: [no] ipv6 mld-snooping version 1 | 2

You can also specify the MLD version for individual VLANs, or individual ports within VLANs. If no MLD version is specified for a VLAN, then the globally configured MLD version is used. If an MLD version is specified for individual ports in a VLAN, those ports use that version instead of the version specified for the VLAN or the globally specified version. The default is MLDv1.

Configuring Report Control

When a device is in passive mode, it forwards reports and leave messages from clients to the upstream router ports that are receiving queries.

You can configure report control to rate-limit report forwarding for the same group to no more than once per 10 seconds. This rate limiting does not apply to the first report answering a group-specific query.

NOTE: This feature applies to MLDv1 only. The leave messages are not rate limited.

MLDv1 membership reports for the same group from different clients are considered to be the same, and are rate-limited. This alleviates the report storm caused by multiple clients answering the upstream router query. To enable report-control, use a command similar to the following:

```
FastIron(config)#ipv6 mld-snooping report-control
```

Syntax: [no] ipv6 mld-snooping report-control

Modifying the Wait Time Before Stopping Traffic When Receiving a Leave Message

You can define the wait time before stopping traffic to a port when the device receives a leave message for that port. The device sends group-specific queries once per second to determine if any client on the same port still needs the group. The value range is from 1 to 5, and the default is 2. Due to the internal timer accuracy, the actual wait time is between n and (n+1) seconds, where n is the configured value.

```
FastIron(config)#ipv6 mld-snooping leave-wait-time 1
```

Syntax: [no] ipv6 mld-snooping leave-wait-time <num>

Modifying the Multicast Cache (mcache) Aging Time

You can set the time for an mcache to age out when it does not receive traffic. The traffic is hardware-switched. One minute before an mcache is aged out, the device mirrors a packet of the mcache to the CPU to reset the age. If no data traffic arrives within one minute, the mcache is deleted. If you configure a lower value, the resource consumed by idle streams is quickly removed, but packets are mirrored to the CPU more frequently. Configure a higher value only when data streams are arriving consistently. The range is 60 to 3600 seconds, and the default is 60 seconds.

```
FastIron Switch(config)#ipv6 mld-snooping mcache-age 180
```

Syntax: [no] ipv6 mld-snooping mcache-age <num>

Disabling Error and Warning Messages

The FGS or FLS device prints error or warning messages when it runs out of software resources or when it receives packets with the wrong checksum or groups. These messages are rate limited. You can turn off these messages by entering a command such as the following:

```
FastIron(config)#ipv6 mld-snooping verbose-off
```

Syntax: [no] ipv6 mld-snooping verbose-off

Configuring the MLD Mode for a VLAN

You can configure a VLAN for either the active or passive (default) MLD mode. The VLAN setting overrides the global setting.

- Active – In active MLD mode, an FGS or FLS device actively sends out MLD queries to identify IPv6 multicast groups on the network, and makes entries in the MLD table based on the group membership reports it receives from the network.
- Passive – In passive MLD mode, the device forwards reports to router ports which receive queries. MLD snooping in the passive mode does not send queries. However, it does forward queries to the entire VLAN.

To set the MLD mode for VLAN 20 to active, enter the following commands:

```
FastIron(config)#vlan 20
FastIron(config-vlan-20)#mld-snooping active
```

Syntax: [no] mld-snooping active | passive

Disabling MLD Snooping for the VLAN

When MLD snooping is enabled globally, you can disable it for a specific VLAN. For example, the following commands disable MLD snooping for VLAN 20. This setting overrides the global setting for VLAN 20.

```
FastIron(config)#vlan 20
FastIron(config-vlan-20)#mld-snooping disable-mld-snoop
```

Syntax: [no] mld-snooping disable-mld-snoop

Configuring the MLD Version for the VLAN

You can specify the MLD version for a VLAN. For example, the following commands configure VLAN 20 to use MLDv2:

```
FastIron(config)#vlan 20
FastIron(config-vlan-20)#mld-snooping version 2
```

Syntax: [no] mld-snooping version 1 | 2

When no MLD version is specified, the globally-configured MLD version is used. If an MLD version is specified for individual ports in the VLAN, these ports use that version, instead of the version specified for the VLAN.

Configuring the MLD Version for Individual Ports in the VLAN

You can specify the MLD version for individual ports in a VLAN. For example, the following commands configure ports 0/1/4, 0/1/5, 0/1/6 and 0/2/1 to use MLDv2. The other ports in the VLAN use the MLD version specified with the **mld-snooping version** command, or the globally configured MLD version.

```
FastIron(config)#vlan 20
FastIron(config-vlan-20)#mld-snooping port-version 2 ethe 0/2/1 ethe 0/1/4 to 0/1/6
```

Syntax: [no] mld-snooping port-version 1 | 2 <port-numbers>

Configuring Static Groups to the Entire VLAN or to Individual Ports

A snooping-enabled VLAN cannot forward multicast traffic to ports that do not receive MLD membership reports. To allow clients to send reports, you can configure a static group which applies to the entire VLAN, or to individual ports on the VLAN. The static group forwards packets to the static group ports even if they have no client membership reports. The static group for the entire VLAN is used in VLAN flooding because it uses fewer hardware resources than the static group for individual ports. Configure a static group for specific ports on VLAN 20 using commands similar to the following:

```
FastIron(config)#vlan 20
FastIron(config-vlan-20)#mld-snooping static-group ff05::100 count 2 ethe 0/1/3 ethe
0/1/5 to 0/1/7
FastIron(config-vlan-20)#mld-snooping static-group ff10::200
```

Syntax: [no] mld-snooping static-group <ipv6-address> [count <num>] [<port-numbers>]

The **ipv6-address** parameter is the IPv6 address of the multicast group.

The **count** is optional, which allows a contiguous range of groups. Omitting the count <num> is equivalent to the count being 1.

If there are no **port-numbers**, the static groups apply to the entire VLAN.

Configuring Static Router Ports

An FGS or FLS device always forwards all multicast control and data packets to router ports that receive queries. Although router ports are learned, you can configure static router ports to force multicast traffic to specific ports, even though these ports never receive queries. To configure static router ports, enter commands such as the following:

```
FastIron Switch(config)#vlan 70

FastIron Switch(config-vlan-70)#mld-snooping router-port e 0/1/4 to 0/1/5 e 0/1/8
```

Syntax: [no] mld-snooping router-port <port-numbers>

Turning off Static Group Proxy

A device with static groups configured acts as a proxy and sends membership reports for its static groups when it receives general or group-specific queries. When a static group configuration is removed, the group is deleted from active group table immediately. However, the device does not send leave messages to the querier. The querier should age the group out. The proxy activity can be turned off (the default is on). For example:

```
FastIron(config)#vlan 20

FastIron(config-vlan-20)#mld-snooping proxy-off
```

Syntax: [no] mld-snooping proxy-off

Enabling MLDv2 Membership Tracking and Fast Leave for the VLAN

MLDv2 provides membership tracking and fast leave services to clients. In MLDv1, only one client per interface must respond to a router's queries; leaving some clients invisible to the router, which makes it impossible for the device to track the membership of all clients in a group. In addition, when a client leaves the group, the device sends group-specific queries to the interface to see if other clients on that interface need the data stream of the client who is leaving. If no client responds, the device waits a few seconds before stopping the traffic. You can configure the wait time with the **ipv6 mld-snooping leave-wait-time** command.

MLDv2 requires that every client respond to queries, allowing the device is able to track every client. When the tracking feature is enabled, the device immediately stops forwarding traffic to the interface if an MLDv2 client sends a leave message, and there is no other client. This feature requires the entire VLAN to be configured for

MLDv2 and have no MLDv1 clients. If a client does not send a report during the specified group membership time (the default is 140 seconds), that client is removed from the tracking list.

Every group on a physical port keeps its own tracking record. However, it can track group membership only; it cannot track by (source, group). For example, Client A and Client B belong to group1 but each is receiving traffic from different sources. Client A receives a traffic stream from (source_1, group1) and Client B receives a traffic stream from (source_2, group1). The device waits for the configured **leave-wait-time** before it stops the traffic because the two clients are in the same group. If the clients are in different groups, the waiting period is ignored and traffic is stopped immediately.

To enable tracking and fast leave for VLAN 20, enter the following commands:

```
FastIron(config)#vlan 20
FastIron(config-vlan-20)#mld-snooping tracking
```

Syntax: [no] mld-snooping tracking

The membership tracking and fast leave features are supported for MLDv2 only. If a port or client is not configured for MLDv2, the **mld-snooping tracking** command is ignored.

Configuring Fast Leave for MLDv1

When an FGS or FLS device receives an MLDv1 leave message, it sends out multiple group-specific queries. If no other client replies within the waiting period, the device stops forwarding traffic to this port. Configuring fast-leave-v1 allows the device to stop forwarding traffic to a port immediately upon receiving a leave message. The device does not send group-specific queries. It is important that no snooping ports have multiple clients. When two devices connect, the querier device should not be configured for fast-leave-v1 because the port to the non-querier device could have multiple clients. The number of queries and the waiting period (in seconds) can be configured using the **ipv6 mld-snooping leave-wait-time** command. The default is 2 seconds. To configure fast leave for MLDv1, use commands such as the following:

```
FastIron(config)#vlan 20
FastIron(config-vlan-20)#mld-snooping fast-leave-v1
```

Syntax: [no] mld-snooping fast-leave-v1

Enabling Fast Convergence

In addition to periodically sending general queries, an active (querier) FGS or FLS device sends out general queries when it detects a new port. However, since it does not recognize the other device's port-up event, the multicast traffic might still use the query-interval time to resume after a topology change. Configuring fast-convergence allows the device to listen to topology change events in L2 protocols, such as spanning tree, and send general queries to shorten the convergence time.

If the L2 protocol is unable to detect a topology change, the fast-convergence feature may not work. For example, if the direct connection between two devices switches from one interface to another, the rapid spanning tree protocol (802.1w) considers this an optimization action, rather than a topology change. In this case, other devices will not receive topology change notifications and will be unable to send queries to speed up the convergence. The original spanning tree protocol does not recognize optimization actions, and fast-convergence works in all cases.

To enable fast-convergence, enter commands such as the following:

```
FastIron Switch(config)#vlan 70

FastIron Switch(config-vlan-70)#mld-snooping fast-convergence
```

Syntax: mld-snooping fast-convergence

Displaying MLD Snooping Information

You can display the following MLD Snooping information:

- MLD Snooping error information
- Information about VLANs

- Group and forwarding information for VLANs
- MLD memory pool usage
- Status of MLD traffic
- MLD information by VLAN

Displaying MLD Snooping Error Information

To display information about possible MLD errors, enter the following command:

```
FastIron#show ipv6 mld-snooping error
snoop SW processed pkt: 173, up-time 160 sec
```

Syntax: show ipv6 mld-snooping error

The following table describes the output from the show ipv6 mld-snooping error command.

This Field...	Displays...
SW processed pkt	The number of IPv6 multicast packets processed by MLD snooping.
up-time	The time since the MLD snooping last occurred is enabled.

Displaying MLD Group Information

To display MLD group information, enter the following command:

```
FastIron Switch#show ipv6 mld-snooping group
p-:physical, ST:static, QR:querier, EX:exclude, IN:include, Y:yes, N:no
VL1 : 263 grp, 263 grp-port, tracking_enabled
      group                                     p-port ST QR life mode source
1     ff0e::ef00:a0e3                           0/1/7  N  Y  120  EX   0
2     ff01::1:f123:f567                         0/1/9  N  Y       IN   1
```

NOTE: In this example, an MLDv1 group is in EXCLUDE mode with a source of 0. The group excludes traffic from the 0 (zero) source list, which actually means that all traffic sources are included.

To display detailed MLD group information, enter the following command:

```
FastIron Switch#show ipv6 mld-snooping group ff0e::ef00:a096 detail
Display group ff0e::ef00:a096 in all interfaces in details.
p-:physical, ST:static, QR:querier, EX:exclude, IN:include, Y:yes, N:no
VL1 : 1 grp, 1 grp-port, tracking_enabled
      group                                     p-port ST QR life mode source
1     ff0e::ef00:a096                           0/1/7  N  Y  100  EX   0
    group: ff0e::ef00:a096, EX, permit 0 (source, life):
      life=100, deny 0:
```

If tracking and fast leave are enabled, you can display the list of clients for a particular group by entering the following command:

```
FastIron Switch#show ipv6 mld-snooping group ff0e::ef00:a096 tracking
Display group ff0e::ef00:a096 in all interfaces with tracking enabled.
p-:physical, ST:static, QR:querier, EX:exclude, IN:include, Y:yes, N:no
VL1 : 1 grp, 1 grp-port, tracking_enabled
      group                                  p-port ST QR life mode source
1     ff0e::ef00:a096                        0/1/7  N  Y  80   EX   0
    receive reports from 1 clients: (age)
      (fe80::1011:1213:1415 60)
```

Syntax: show ipv6 mld-snooping group [<group-address> [detail] [tracking]]

To receive a report for a specific multicast group, enter that group's address for <group-address>.

Enter **detail** to display the source list of a specific VLAN.

Enter **tracking** for information on interfaces that are tracking-enabled.

The following table describes the information displayed by the **show ipv6 mld-snooping group** command.

This Field...	Displays...
group	The address of the IPv6 group (destination IPv6 address).
p-port	The physical port on which the group membership was received.
ST	**Yes** indicates that the MLD group was configured as a static group; **No** means it was learned from reports.
QR	**Yes** means the port is a querier port; **No** means it is not. A port becomes a non-querier port when it receives a query from a source with a lower source IP address than the port.
life	The number of seconds the group can remain in EXCLUDE mode. An EXCLUDE mode changes to INCLUDE if it does not receive an IS_EX or TO_EX message during a specified period of time. The default is 140 seconds. There is no `life` displayed in INCLUDE mode.
mode	The current mode of the interface: INCLUDE or EXCLUDE. If the interface is in INCLUDE mode, it admits traffic only from the source list. If the interface is in EXCLUDE mode, it denies traffic from the source list and accepts the rest.
source	Identifies the source list that will be included or excluded on the interface. An MLDv1 group is in EXCLUDE mode with a source of 0. The group excludes traffic from 0 (zero) source list, which actually means that all traffic sources are included.
group	If you requested a *detailed* report, the following information is displayed: • The multicast group address • The mode of the group • Sources from which traffic will be admitted (INCLUDE) or denied (EXCLUDE) on the interface. • The life of each source list. If you requested a *tracking/fast leave* report, the clients from which reports were received are identified.

Displaying MLD Snooping Mcache Information

The MLD snooping mcache contains multicast forwarding information for VLANs. To display information in the multicast forwarding mcache, enter the following command:

```
FastIron Switch#show ipv6 mld-snooping mcache
Example: (S G) cnt=: (S G) are the lowest 32 bits, cnt: SW proc. count
    OIF: 0/1/22 TR(0/1/32,0/1/33), TR is trunk, 0/1/32 primary, 0/1/33 output
vlan 1, has 2 cache
1    (abcd:ef50 0:100), cnt=121
     OIF: 0/1/11 0/1/9
     age=0s up-time=120m vidx=4130 (ref-cnt=1)
2    (abcd:ef50 0:101), cnt=0
     OIF: entire vlan
     age=0s up-time=0m vidx=8191 (ref-cnt=1)
vlan 70, has 0 cache
```

Syntax: show ipv6 mld-snooping mcache

The following table describes the output from the **ipv6 mld-snooping mcache** command.

This Field...	DIsplays...
(abcd:ef50 0:100):	The lowest 32 bits of source and group. It is displayed in XXXX:XXXX hex format. Here XXXX is a 16-bit hex number.
cnt	The number of packets processed in software. IPv6 packets are switched in software in the FGS 03.0.00 release causing this number to increase slowly.
OIF	Output interfaces. *Entire vlan* means that static groups apply to the entire VLAN.
age	The mcache age. The mcache is reset to 0 if traffic continues to arrive, otherwise it is aged out when it reaches the time defined by **ipv6 mld-snooping mcache-age**.
uptime	The up time of this mcache in minutes.
vidx	The vidx is shared among mcaches using the same output interfaces. The vidx specifies the output port list, which shows the index. Valid range is from 4096 to 8191.
ref-cnt	The number of mcaches using this vidx.

Displaying Software Resource Usage for VLANs

To display information about the software resources used, enter the following command:

```
FastIron Switch#show ipv6 mld-snooping resource
                      alloc in-use  avail get-fail    limit  get-mem  size init
mld group               512      9    503        0    32000      272    28  256
mld phy port           1024     16   1008        0   200000      279    21 1024
snoop group hash        512      9    503        0    59392      272    20  256
…. Entries deleted
total pool memory 194432 bytes
has total 1 forwarding hash
Available vidx: 4061
```

Syntax: show ipv6 mld-snooping resource

The following table describes the output from the show **ipv6 mld-snooping resource** command:

This Field...	**Displays...**
alloc	The allocated number of units.
in-use	The number of units which are currently used.
avail	The number of available units.
get-fail	Displays the number of resource failures. **NOTE:** It is important to pay close attention to this field.
limit	The upper limit of this expandable field. The MLD group limit is configured using the **system-max mld-max-group-addr** command. The snoop mcache entry limit is configured using the **system-max mld-snoop-mcache** command.
get-mem	The number of memory allocation. This number should continue to increase.
size	The size of a unit (in bytes).
init	The initial allocated amount of memory. **NOTE:** This number can be increased. More memory can be allocated if necessary.
Available vidx	The output interface (OIF) port mask used by mcache. The entire device has a maximum of 4096 vidx. Different mcaches with the same OIF share the same vidx. If vidx is not available, the stream cannot be hardware-switched.

Displaying Status of MLD Snooping Traffic

To display status information for MLD snooping traffic, enter the following command:

```
FastIron Switch#show ipv6 mld-snooping traffic

MLD snooping: Total Recv: 32208, Xmit: 166
Q: query, Qry: general Q,  G-Qry: group Q,  GSQry: group-source Q, Mbr: member
Recv      QryV1     QryV2     G-Qry     GSQry     MbrV1     MbrV2     Leave
VL1           0         0         0         0     31744       208       256
VL70          0         0         0         0         0         0         0
Recv       IsIN      IsEX      ToIN      ToEX     ALLOW     BLOCK   Pkt-Err
VL1        1473     31784         0         1         1         7         0
VL70          0         0         0         0         0         0         0
Send      QryV1     QryV2     G-Qry     GSQry     MbrV1     MbrV2
VL1           0         0       166         0         0         0
VL70          0         0         0         0         0         0
```

Syntax: show ipv6 mld-snooping traffic

The following table describes the information displayed by the **show ipv6 mld-snooping traffic** command.

This Field	**Displays**
Q	Query
Qry	General Query

This Field	Displays
QryV1	Number of general MLDv1 queries received or sent.
QryV2	Number of general MLDv2 snooping queries received or sent.
G-Qry	Number of group specific queries received or sent.
GSQry	Number of group source specific queries received or sent.
MBR	The membership report.
MbrV1	The MLDv1 membership report.
MbrV2	The MLDv2 membership report.
IsIN	Number of source addresses that were included in the traffic.
IsEX	Number of source addresses that were excluded in the traffic.
ToIN	Number of times the interface mode changed from EXCLUDE to INCLUDE.
ToEX	Number of times the interface mode changed from INCLUDE to EXCLUDE.
ALLO	Number of times additional source addresses were allowed on the interface.
BLK	Number of times sources were removed from an interface.
Pkt-Err	Number of packets having errors such as checksum errors.

Displaying MLD Snooping Information by VLAN

You can display MLD snooping information for all VLANs or for a specific VLAN. For example, to display MLD snooping information for VLAN 70, enter the following command:

```
FastIron Switch#show ipv6 mld-snooping vlan 70
version=1, query-t=60, group-aging-t=140, max-resp-t=3, other-qr-present-t=123
VL70: cfg V2, vlan cfg passive, 2 grp, 0 (SG) cache, rtr ports,
    router ports: 0/1/36(120) fe80::2e0:52ff:fe00:9900,
  0/1/26 has 2 grp, non-QR (passive), cfg V1
  0/1/26 has 2 grp, non-QR (passive), cfg V1
    group: ff10:1234::5679, life = 100
    group: ff10:1234::5678, life = 100
  0/1/35 has 0 grp, non-QR (QR=fe80::2e0:52ff:fe00:9900, age=20), dft V2 trunk
```

Syntax: show ipv6 mld-snooping vlan [<vlan-id>]

If you do not specify a vlan-id, information for all VLANs is displayed.

The following table describes information displayed by the **show ipv6 mld-snooping vlan** command.

This Field	Displays
version	The MLD version number.
query-t	How often a querier sends a general query on the interface.
group-aging-t	Number of seconds membership groups can be members of this group before aging out.

This Field	Displays
rtr-port	The router ports which are the ports receiving queries. The display `router ports: 0/1/36(120) fe80::2e0:52ff:fe00:9900` means port 0/1/36 has a querier with fe80::2e0:52ff:fe00:9900 as the link-local address, and the remaining life is 120 seconds.
max-resp-t	The maximum number of seconds a client can wait before it replies to the query.
non-QR	Indicates that the port is a non-querier.
QR	Indicates that the port is a querier.

Clear MLD Snooping Commands

The clear commands for MLD snooping should only be used in troubleshooting situations or when recovering from error conditions.

Clear MLD Counters on VLANs

To clear MLD Snooping error and traffic counters on all VLANs, enter a command similar to the following:

```
FastIron#clear ipv6 mld-snooping counters
```

Syntax: clear ipv6 mld-snooping counters

Clear MLD mcache

To clear the mcache on all VLANs, enter the following command:

```
FastIron#clear ipv6 mld-snooping mcache
```

Syntax: clear ipv6 mld-snooping mcache

Clear mcache on a Specific VLAN

To clear the mcache on a specific VLAN, enter the following command:

```
FastIron#clear ipv6 mld-snooping vlan 10 mcache
```

Syntax: clear ipv6 mld-snooping vlan <vlan-id> mcache

The <vlan-id> parameter specifies the specific VLAN from which to clear the cache.

Clear Traffic on a Specific VLAN

To clear the traffic counters on a specific VLAN, enter the following command:

```
FastIron#clear ipv6 mld-snooping vlan 10 traffic
```

Syntax: clear ipv6 mld-snooping vlan <vlan-id> traffic

The <vlan-id> parameter specifies the specific VLAN from which to clear the traffic counters.

Chapter 32

Configuring Multicast Listening Discovery (MLD) Snooping on the FastIron X Series Switch

This chapter describes how to configure Multicast Listening Discovery (MLD) Snooping on Foundry FESX and FSX devices running software release 04.1.00 and later (devices running IPv6).

Platform Support:

- FESX and FSX IPv6 devices running software release 04.1.00 and later - L2, BL3, L3

Overview

The default method a FastIron uses to process an IPv6 multicast packet is to broadcast it to all ports except the incoming port of a VLAN. Packets are flooded by hardware without going to the CPU, which may result in some clients receiving unwanted traffic.

If a VLAN is not MLD snooping-enabled, it floods IPv6 multicast data and control packets to the entire VLAN in hardware. When snooping is enabled, MLD packets are trapped to the CPU. Data packets are mirrored to the CPU and flooded to the entire VLAN. The CPU then installs hardware resources so subsequent data packets can be hardware-switched to desired ports without going through the CPU. If there is no client report, the hardware resource drops the data stream.

MLD protocols provide a way for clients and a device to exchange messages, and allow the device to build a database indicating which port wants what traffic. Since the MLD protocols do not specify forwarding methods, MLD snooping or multicast protocols such as IPv6 PIM-Sparse Mode (PIM SM) are required to handle packet forwarding. PIM SM can route multicast packets within and outside a VLAN, while MLD snooping can switch packets only within a VLAN. FESX and FSX devices do not support PIM-SM routing.

MLD snooping provides multicast containment by forwarding traffic only to those clients that have MLD receivers for a specific multicast group (destination address). The FastIron maintains the MLD group membership information by processing MLD reports and generating messages so traffic can be forwarded to ports receiving MLD reports. This is analogous to IGMP Snooping on Foundry Layer 3 switches.

An IPv6 multicast address is a destination address in the range of FF00::/8. A limited number of multicast addresses are reserved. Since packets destined for the reserved addresses may require VLAN flooding, FESX and FSX devices don't snoop in the FF0X::000X range (where X is from 0 to F). Data packets destined to these addresses are flooded to the entire VLAN by hardware and mirrored to the CPU. Multicast data packets destined to addresses outside the FF0X::000X range are snooped. A client must send MLD reports in order to receive traffic.

An MLD device periodically broadcasts general queries and sends group queries upon receiving a leave message, to ensure no other clients at the same port still want this specific traffic before removing it. MLDv1 allows clients to specify which group (destination IPv6 address) will receive traffic. (MLDv1 cannot choose the source of the traffic.) MLDv2 deals with source-specific multicasts, adding the capability for clients to INCLUDE or EXCLUDE specific traffic sources. An MLDv2 device's port state can either be in INCLUDE or EXCLUDE mode.

There are different types of group records for client reports. Clients respond to general queries by sending a membership report containing one or more of the following records associated with a specific group:

- Current-state record - Indicates the sources from which the client wants to receive or not receive traffic. This record contains the source addresses of the clients and indicates whether or not traffic will be included (IS_IN) or excluded (IS_EX) from that source address.
- Filter-mode-change record - If the client changes its current state from IS_IN to IS_EX, a TO_EX record is included in the membership report. Likewise, if a client's current state changes from IS_EX to IS_IN, a TO_IN record appears in the membership report.
- MLDv1 leave report - Equivalent to a TO_IN (empty) record in MLDv2. This record means that no traffic from this group will be received, regardless of the source.
- An MLDv1 group report - Equivalent to an IS_EX (empty) record in MLDv2. This record means that all traffic from this group will be received, regardless of the source.
- Source-list-change record - If the client wants to add or remove traffic sources from its membership report, the report can include an ALLOW record, which contains a list of new sources from which the client wishes to receive traffic. The report can also contain a BLOCK record, which lists current traffic sources from which the client wants to stop receiving traffic.

How MLD Snooping Uses MAC Addresses to Forward Multicast Packets

MLD snooping on FastIron devices is based on MAC address entries. When an IPv6 multicast data packet is received, the packet's destination MAC is matched with the MAC address entries in the IPv6 multicast table. If a match is found, packets are sent to the ports associated with the MAC address. If a match is not found, packets are flooded to the VLAN and copied to the CPU.

For IPv6 multicast, the destination MAC address is in the format 0x33-33-xx-yy-zz-kk, where xx-yy-zz-kk are the 32 lowest bits of the IPv6 multicast group address. For example, the IPv6 group address 0xFF3E:40:2001:660:3007:123:1234:5678 maps to the IPv6 MAC address 0x33-33-12-34-56-78.

For two multicast traffic streams, Source_1 and Group1 (S1,G1) and Source_2 and Group2 (S2,G2), with the same or different source addresses, if the lowest 32 bits of the 128-bit IPv6 group address are the same, they would map to the same destination MAC. Because the FESX and FSX support MAC-based forwarding for MLD snooping, the final multicast MAC address entry would be a superset of all the IPv6 groups mapped to it. For example, consider the following three IPv6 multicast streams sent from port 5 of a FastIron device:

(S1,G1) = (2060::5, ff1e::12), client port 1, port 2

(S2,G2) = (2060::6, ff2e::12), client port 2, port 3

(S3,G1) = (2060::7, ff1e::12), client port 4

Because the lowest 32 bits of the group address for G1 and G2 are the same, all three streams would use 0x33-33-00-00-00-12 as the destination MAC address. MLD snooping would build a MAC entry with the MAC address 0x33-33-00-00-00-12 on egress ports 1, 2, 3, and 4. As a result, all three streams would be sent to ports 1, 2, 3, and 4. Note that the above example assumes the following:

- The FastIron device is running MLD snooping on VLAN 10 and all three streams are in VLAN 10
- There are clients on port 1 and port 2 for (S1,G1)
- There are clients on port 2 and port 3 for (S2,G2)
- There are clients on port 4 for (S3,G1)

Configuration Notes

- Servers (traffic sources) are not required to send MLD memberships.
- The default MLD version is V1, where the source address is not sensitive. In the example given in the preceding section ("How MLD Snooping Uses MAC Addresses to Forward Multicast Packets" on page 32-2), (S1,G1) and (S3,G1) would be considered the same group as (*,G1).

- If MLDv2 is configured on any port of a VLAN, you can check the source information, but because MLD snooping is MAC based, (S,G) switching is not feasible.
- Hardware resources are installed only when there is data traffic.
- The FESX and FSX support up to 4K collective entries for IGMP Snooping, MLD snooping, and static multicast MAC addresses.
- You can configure the maximum number of groups and the multicast cache (mcache) number.
- The device supports static groups applying to specific ports. The device acts as a proxy to send MLD reports for the static groups when receiving queries.
- A user can configure static router ports, forcing all multicast traffic to be sent to these ports.
- Foundry FastIron devices support fast leave for MLDv1, which stops traffic immediately to any port that has received a leave message.
- Foundry FastIron devices support tracking and fast leave for MLDv2, which tracks all MLDv2 clients. If the only client on a port leaves, traffic is stopped immediately.
- An MLD device can be configured as a querier (active) or non-querier (passive). Queriers send queries. Non-queriers listen for queries and forward them to the entire VLAN.
- Every VLAN can be independently configured as a querier or a non-querier.
- A VLAN that has a connection to an IPv6 PIM-enabled port on another router should be configured as a non-querier. When multiple snooping devices connect together and there is no connection to IPv6 PIM ports, only one device should be configured as the querier. If multiple devices are configured as active, only one will continue to send queries after the devices have exchanged queries. See “Queriers and Non-Queriers” on page 32-3.
- An MLD device can be configured to rate-limit the forwarding of MLDv1 membership reports to queriers.
- Because FESX and FSX devices use an IPv6 link-local address as the source address when sending queries, no global address is required.
- The MLD implementation allows snooping on some VLANs or on all VLANs. MLD can be enabled or disabled independently for each VLAN. In addition, individual ports of a VLAN can be configured as MLDv1 and MLDv2. In general, global configuration commands such as **ipv6 mld-snooping..** apply to all VLANs except those with a local **mld-snooping..** configuration, which supersedes the global configuration. Configuring the version on a port or a VLAN only affects the device’s sent query version. The device always processes all versions of client reports regardless of the version configured.
- MLD snooping requires hardware resources. If the device has insufficient resources, the data stream without a resource is mirrored to the CPU in addition to being VLAN flooded, which can cause high CPU usage. To avoid this situation, Foundry recommends that you avoid enabling snooping globally unless necessary.
- To receive data traffic, MLD snooping requires clients to send membership reports. If a client does not send reports, you must configure a static group to force traffic to client ports.

Queriers and Non-Queriers

An MLD snooping-enabled device can be configured as a querier (active) or non-querier (passive). An MLD querier sends queries; a non-querier listens for MLD queries and forwards them to the entire VLAN. When multiple MLD snooping devices are connected together, and there is no connection to an IPv6 PIM-enabled port, one of the devices should be configured as a querier. If multiple devices are configured as queriers, after multiple devices exchange queries, then all devices except the winner (the device with the lowest address) stop sending queries. Although the system works when multiple devices are configured as queriers, Foundry recommends that only one device, preferably the one with the traffic source, is configured as the querier.

VLANs can also be independently configured as queriers or non-queriers. If a VLAN has a connection to an IPv6 PIM-enabled port on another router, the VLAN should be configured as a non-querier.

Because non-queriers always forward multicast data traffic and MLD messages to router ports which receive MLD queries or IPv6 PIM hellos, Foundry recommends that you configure the devices with the data traffic source (server) as queriers. If a server is attached to a non-querier, the non-querier always forwards traffic to the querier regardless of whether or not there are clients on the querier.

NOTE: In a topology with one or more connected devices, at least one device must be running PIM, or configured as active. Otherwise, no devices can send queries, and traffic cannot be forwarded to clients.

To configure the MLD mode (querier or non-querier) on an MLD snooping-enabled device, see “Configuring the Global MLD Mode” on page 32-5. To configure the MLD mode on a VLAN, see “Configuring the MLD Mode for a VLAN” on page 32-7.

VLAN Specific Configuration

You can configure MLD snooping on some VLANs or all VLANs. Each VLAN can be independently enabled or disabled for MLD snooping, or can be configured with MLDv1 or MLDv2. In general, the **ipv6 mld-snooping...** commands apply globally to all VLANs except those configured with VLAN-specific **mld-snooping...** commands. VLAN-specific **mld-snooping** commands supersede global **ipv6 mld-snooping** commands.

Using MLDv1 with MLDv2

MLD snooping can be configured as MLDv1 or MLDv2 on individual ports on a VLAN. An interface or router sends queries and reports that include the MLD version with which it has been configured. The version configuration applies only to the sending of queries. The snooping device recognizes and processes MLDv1 and MLDv2 packets regardless of the version configured.

NOTE: To avoid version deadlock, when an interface receives a report with a lower version than that for which it has been configured, the interface does **not** automatically downgrade the running MLD version.

Configuring MLD Snooping

Configuring MLD snooping on an IPv6 device consists of the following global and VLAN-specific tasks:

Global Tasks:

- Configuring hardware and software resource limits
- Disabling transmission and receipt of MLD packets on a port
- Configuring the MLD mode: active or passive (must be enabled for MLD snooping)
- Modifying the age interval
- Modifying the interval for query messages (active MLD mode only)
- Specifying the global MLD version
- Enabling and disabling report control (rate limiting)
- Modifying the leave wait time
- Modifying the mcache age interval
- Disabling error and warning messages

VLAN-Specific Tasks:

- Configuring the MLD mode for the VLAN: active or passive
- Enabling or disabling MLD snooping for the VLAN
- Configuring the MLD version for the VLAN
- Configuring the MLD version for individual ports in the VLAN
- Configuring static groups to the entire VLAN or some ports

- Configuring static router ports
- Disabling proxy activity for a static group
- Enabling client tracking and the fast leave feature for MLDv2
- Configuring fast leave for MLDv1
- Configuring fast-convergence

Configuring the Hardware and Software Resource Limits

The system supports up to 8K of hardware-switched multicast streams. The configurable range is from 256 to 8192 and the default is 4096. To define the maximum number of MLD snooping mcache entries, enter a command such as the following:

```
FastIron(config)#system-max mld-snoop-mcache 8000
```

Syntax: [no] system-max mld-snoop-mcache <num>

The system supports up to 32K of groups. The configurable range is 256 to 32768 and the default is 8192. The configured number is the upper limit of an expandable database. Client memberships exceeding the group limits are not processed.

Disabling Transmission and Receipt of MLD packets on a Port

When a VLAN is snooping-enabled, all MLD packets are trapped to the CPU without hardware VLAN flooding. The CPU can block MLD packets to and from a multicast-disabled port, and will not add that port to the output interfaces of hardware resources, which prevents the disabled port from receiving multicast traffic. However, if static groups to the entire VLAN are defined, the traffic for these groups is flooded to the entire VLAN, including to the disabled ports. Since the hardware cannot block traffic from disabled ports, hardware traffic is switched in the same way as traffic from enabled ports.

NOTE: This command has no effect on a VLAN that is not snooping-enabled because all multicast traffic is VLAN flooded.

```
FastIron(config)#interface ethernet 1/3
FastIron(config-if-e1000-1/3)#ipv6-multicast-disable
```

Syntax: [no] ipv6-multicast-disable

Configuring the Global MLD Mode

You can configure a Foundry FastIron device for either active or passive (default) MLD mode. If you specify an MLD mode for a VLAN, the MLD mode overrides the global setting.

- Active – In active MLD mode, a device actively sends out MLD queries to identify IPv6 multicast groups on the network, and makes entries in the MLD table based on the group membership reports it receives from the network.
- Passive – In passive MLD mode, the device forwards reports to the router ports which receive queries. MLD snooping in passive mode does not send queries, but does forward queries to the entire VLAN.

To globally set the MLD mode to active, enter the following command:

```
FastIron(config)#ipv6 mld-snooping active
```

Syntax: [no] ipv6 mld-snooping [active | passive]

Omitting both the **active** and **passive** keywords is the same as entering **ipv6 mld-snooping passive**.

Modifying the Age Interval

When the device receives a group membership report, it makes an entry in the MLD group table for the group in the report. The age interval specifies how long the entry can remain in the table without the device receiving another group membership report. When multiple devices connect together, all devices should be configured with the same age interval. The age interval should be at least twice that of the query interval, so that missing one report won't stop traffic. For a non-querier, the query interval should equal that of the querier.

To modify the age interval, enter a command such as the following:

```
FastIron(config)#ipv6 mld-snooping age-interval 280
```

Syntax: [no] ipv6 mld-snooping age-interval <interval>

The <interval> parameter specifies the aging time. You can specify a value from 20 – 7200 seconds. The default is 140 seconds.

Modifying the Query Interval (Active MLD Snooping Mode Only)

If the MLD mode is set to active, you can modify the query interval, which specifies how often the FastIron device sends group membership queries. By default, queries are sent every 60 seconds. When multiple queriers connect together, all queriers should be configured with the same interval.

To modify the query interval, enter a command such as the following:

```
FastIron(config)#ipv6 mld-snooping query-interval 120
```

Syntax: [no] ipv6 mld-snooping query-interval <interval>

The <interval> parameter specifies the interval between queries. You can specify a value from 10 – 3600 seconds. The default is 60 seconds.

Configuring the Global MLD Version

The default version is MLDv1. You can specify the global MLD version on the device as either MLDv1 or MLDv2. For example, the following command configures the device to use MLDv2:

```
FastIron(config)#ipv6 mld-snooping version 2
```

Syntax: [no] ipv6 mld-snooping version 1 | 2

You can also specify the MLD version for individual VLANs, or individual ports within VLANs. If no MLD version is specified for a VLAN, then the globally configured MLD version is used. If an MLD version is specified for individual ports in a VLAN, those ports use that version instead of the version specified for the VLAN or the globally specified version. The default is MLDv1.

Configuring Report Control

When a device is in passive mode, it forwards reports and leave messages from clients to the upstream router ports that are receiving queries.

You can configure report control to rate-limit report forwarding for the same group to no more than once per 10 seconds. This rate limiting does not apply to the first report answering a group-specific query.

NOTE: This feature applies to MLDv1 only. The leave messages are not rate limited.

MLDv1 membership reports for the same group from different clients are considered to be the same, and are rate-limited. This alleviates the report storm caused by multiple clients answering the upstream router query.

To enable report-control, enter the following command:

```
FastIron(config)#ipv6 mld-snooping report-control
```

Syntax: [no] ipv6 mld-snooping report-control

Modifying the Wait Time Before Stopping Traffic When Receiving a Leave Message

You can define the wait time before stopping traffic to a port when the device receives a leave message for that port. The device sends group-specific queries once per second to determine if any client on the same port still needs the group.

```
FastIron(config)#ipv6 mld-snooping leave-wait-time 1
```

Syntax: [no] ipv6 mld-snooping leave-wait-time <num>

where <num> is a value from 1 to 5. The default is 2. Because of the internal timer accuracy, the actual wait time is between n and (n+1) seconds, where n is the configured value.

Modifying the Multicast Cache (mcache) Aging Time

You can set the time for an mcache to age out when it does not receive traffic. Two seconds before an mcache is aged out, the device mirrors a packet of the mcache to the CPU to reset the age. If no data traffic arrives within two seconds, the mcache is deleted. Note that in FESX and FSX devices, more than one mcache can be mapped to the same destination MAC. Hence, when an mcache entry is deleted, the MAC entry may not be deleted. If you configure a lower value, the resource consumed by idle streams is quickly removed, but packets are mirrored to the CPU more frequently. Configure a higher value only when data streams are arriving consistently.

You can use the **show ipv6 mld-snooping mcache** command to view the currently configured mcache age. See "Displaying MLD Snooping Mcache Information" on page 32-12.

To modify the multicast cache age out time, enter a command such as the following:

```
FastIron(config)#ipv6 mld-snooping mcache-age 180
```

Syntax: [no] ipv6 mld-snooping mcache-age <num>

where <num> is a value from 60 to 3600 seconds, and the default is 60 seconds.

Disabling Error and Warning Messages

The FastIron device prints error or warning messages when it runs out of software resources or when it receives packets with the wrong checksum or groups. These messages are rate limited. You can turn off these messages by entering a command such as the following:

```
FastIron(config)#ipv6 mld-snooping verbose-off
```

Syntax: [no] ipv6 mld-snooping verbose-off

Configuring the MLD Mode for a VLAN

You can configure a VLAN for either the active or passive (default) MLD mode. The VLAN setting overrides the global setting.

- Active – In active MLD mode, a FastIron device actively sends out MLD queries to identify IPv6 multicast groups on the network, and makes entries in the MLD table based on the group membership reports it receives from the network.
- Passive – In passive MLD mode, the device forwards reports to router ports that receive queries. MLD snooping in the passive mode does not send queries. However, it does forward queries to the entire VLAN.

To set the MLD mode for VLAN 20 to active, enter the following commands:

```
FastIron(config)#vlan 20
FastIron(config-vlan-20)#mld-snooping active
```

Syntax: [no] mld-snooping active | passive

The default mode is passive.

Disabling MLD Snooping for the VLAN

When MLD snooping is enabled globally, you can disable it for a specific VLAN. For example, the following commands disable MLD snooping for VLAN 20. This setting overrides the global setting for VLAN 20.

```
FastIron(config)#vlan 20
FastIron(config-vlan-20)#mld-snooping disable-mld-snoop
```

Syntax: [no] mld-snooping disable-mld-snoop

Configuring the MLD Version for the VLAN

You can specify the MLD version for a VLAN. For example, the following commands configure VLAN 20 to use MLDv2:

```
FastIron(config)#vlan 20
FastIron(config-vlan-20)#mld-snooping version 2
```

Syntax: [no] mld-snooping version 1 | 2

When no MLD version is specified, the globally-configured MLD version is used. If an MLD version is specified for individual ports in the VLAN, these ports use that version, instead of the version specified for the VLAN.

Configuring the MLD Version for Individual Ports in the VLAN

You can specify the MLD version for individual ports in a VLAN. For example, the following commands configure ports 1/4, 1/5, 1/6 and 2/1 to use MLDv2. The other ports in the VLAN use the MLD version specified with the **mld-snooping version** command, or the globally configured MLD version.

```
FastIron(config)#vlan 20
FastIron(config-vlan-20)#mld-snooping port-version 2 ethe 2/1 ethe 1/4 to 1/6
```

Syntax: [no] mld-snooping port-version 1 | 2 ethernet <port-numbers>

Configuring Static Groups to the Entire VLAN or to Individual Ports

A snooping-enabled VLAN cannot forward multicast traffic to ports that do not receive MLD membership reports. To allow clients to send reports, you can configure a static group which applies to the entire VLAN, or to individual ports on the VLAN. The static group forwards packets to the static group ports even if they have no client membership reports. The static group for the entire VLAN is used in VLAN flooding because it uses fewer hardware resources than the static group for individual ports. Configure a static group for specific ports on VLAN 20 using commands similar to the following:

```
FastIron(config)#vlan 20
FastIron(config-vlan-20)#mld-snooping static-group ff05::100 count 2 ethe 1/3 ethe
1/5 to 1/7
FastIron(config-vlan-20)#mld-snooping static-group ff10::200
```

Syntax: [no] mld-snooping static-group <ipv6-address> [count <num>] [<port-numbers>]

The **ipv6-address** parameter is the IPv6 address of the multicast group.

The **count** is optional, which allows a contiguous range of groups. Omitting the count <num> is equivalent to the count being 1.

If there are no **port-numbers**, the static groups apply to the entire VLAN.

Configuring Static Router Ports

A FastIron device always forwards all multicast control and data packets to router ports that receive queries. Although router ports are learned, you can configure static router ports to force multicast traffic to specific ports, even though these ports never receive queries. To configure static router ports, enter commands such as the following:

```
FastIron(config)#vlan 70

FastIron(config-vlan-70)#mld-snooping router-port e 1/4 to 1/5 e 1/8
```

Syntax: [no] mld-snooping router-port ethernet <port-numbers>

Disabling Static Group Proxy

A device with statically configured groups acts as a proxy and sends membership reports for its static groups when it receives general or group-specific queries. When a static group configuration is removed, the group is immediately deleted from the active group table. However, the device does not send leave messages to the querier. The querier should age out the group. The proxy activity can be disabled (the default is enabled). For example:

```
FastIron(config)#vlan 20
FastIron(config-vlan-20)#mld-snooping proxy-off
```

Syntax: [no] mld-snooping proxy-off

By default, MLD snooping proxy is enabled.

Enabling MLDv2 Membership Tracking and Fast Leave for the VLAN

MLDv2 provides membership tracking and fast leave services to clients. In MLDv1, only one client per interface must respond to a router's queries; leaving some clients invisible to the router, which makes it impossible for the device to track the membership of all clients in a group. In addition, when a client leaves the group, the device sends group-specific queries to the interface to see if other clients on that interface need the data stream of the client who is leaving. If no client responds, the device waits a few seconds before stopping the traffic. You can configure the wait time with the **ipv6 mld-snooping leave-wait-time** command. See "Modifying the Wait Time Before Stopping Traffic When Receiving a Leave Message" on page 32-7.

MLDv2 requires that every client respond to queries, allowing the device to track every client. When the tracking feature is enabled, the device immediately stops forwarding traffic to the interface if an MLDv2 client sends a leave message, and there is no other client. This feature requires the entire VLAN to be configured for MLDv2 and have no MLDv1 clients. If a client does not send a report during the specified group membership time (the default is 140 seconds), that client is removed from the tracking list.

Every group on a physical port keeps its own tracking record. However, it can track group membership only; it cannot track by (source, group). For example, Client A and Client B belong to group1 but each is receiving traffic from different sources. Client A receives a traffic stream from (source_1, group1) and Client B receives a traffic stream from (source_2, group1). The device waits for the configured **leave-wait-time** before it stops the traffic because the two clients are in the same group. If the clients are in different groups, the waiting period is ignored and traffic is stopped immediately.

To enable tracking and fast leave for VLAN 20, enter the following commands:

```
FastIron(config)#vlan 20
FastIron(config-vlan-20)#mld-snooping tracking
```

Syntax: [no] mld-snooping tracking

The membership tracking and fast leave features are supported for MLDv2 only. If a port or client is not configured for MLDv2, the **mld-snooping tracking** command is ignored.

Configuring Fast Leave for MLDv1

When a FESX or FSX device receives an MLDv1 leave message, it sends out multiple group-specific queries. If no other client replies within the waiting period, the device stops forwarding traffic to this port. Configuring fast-leave-v1 allows the device to stop forwarding traffic to a port immediately upon receiving a leave message. The device does not send group-specific queries. It is important that no snooping ports have multiple clients. When two devices connect, the querier device should not be configured for fast-leave-v1 because the port to the non-querier device could have multiple clients. The number of queries and the waiting period (in seconds) can be configured using the **ipv6 mld-snooping leave-wait-time** command. See "Modifying the Wait Time Before Stopping Traffic When Receiving a Leave Message" on page 32-7.

To configure fast leave for MLDv1, use commands such as the following:

```
FastIron(config)#vlan 20
FastIron(config-vlan-20)#mld-snooping fast-leave-v1
```

Syntax: [no] mld-snooping fast-leave-v1

Enabling Fast Convergence

In addition to periodically sending general queries, an active (querier) FESX or FSX device sends out general queries when it detects a new port. However, since it does not recognize the other device's port-up event, the multicast traffic might still use the query-interval time to resume after a topology change. Configuring fast-convergence allows the device to listen to topology change events in Layer 2 (L2) protocols, such as spanning tree, and send general queries to shorten the convergence time.

If the L2 protocol is unable to detect a topology change, the fast-convergence feature may not work. For example, if the direct connection between two devices switches from one interface to another, the rapid spanning tree protocol (802.1w) considers this an optimization action, rather than a topology change. In this case, other devices will not receive topology change notifications and will be unable to send queries to speed up the convergence. The original spanning tree protocol does not recognize optimization actions, and fast-convergence works in all cases.

To enable fast-convergence, enter commands such as the following:

```
FastIron(config)#vlan 70
FastIron(config-vlan-70)#mld-snooping fast-convergence
```

Syntax: [no] mld-snooping fast-convergence

Displaying MLD Snooping Information

You can display the following MLD snooping information:

- MLD snooping error information
- Group and forwarding information for VLANs
- Information about MLD snooping mcache
- MLD memory pool usage
- Status of MLD traffic
- MLD information by VLAN

Displaying MLD Snooping Error Information

To display information about possible MLD errors, enter the following command:

```
FastIron#show ipv6 mld-snooping error

snoop SW processed pkt: 173, up-time 160 sec
```

Syntax: show ipv6 mld-snooping error

The following table describes the output from the **show ipv6 mld-snooping error** command.

This Field...	Displays...
SW processed pkt	The number of IPv6 multicast packets processed by MLD snooping.
up-time	The MLD snooping up time.

Displaying MLD Group Information

To display MLD group information, enter the following command:

```
FastIron#show ipv6 mld-snooping group
p-:physical, ST:static, QR:querier, EX:exclude, IN:include, Y:yes, N:no
VL1 : 263 grp, 263 grp-port, tracking_enabled
      group                                 p-port ST QR life mode source
1     ff0e::ef00:a0e3                       1/7    N  Y  120  EX   0
2     ff01::1:f123:f567                     1/9    N  Y       IN   1
```

NOTE: In this example, an MLDv1 group is in EXCLUDE mode with a source of 0. The group excludes traffic from the 0 (zero) source list, which actually means that all traffic sources are included.

To display detailed MLD group information, enter the following command:

```
FastIron#show ipv6 mld-snooping group ff0e::ef00:a096 detail
Display group ff0e::ef00:a096 in all interfaces in details.
p-:physical, ST:static, QR:querier, EX:exclude, IN:include, Y:yes, N:no
VL1 : 1 grp, 1 grp-port, tracking_enabled
      group                                 p-port ST QR life mode source
1     ff0e::ef00:a096                       1/7    N  Y  100  EX   0
    group: ff0e::ef00:a096, EX, permit 0 (source, life):
      life=100, deny 0:
```

If tracking and fast leave are enabled, you can display the list of clients for a particular group by entering the following command:

```
FastIron#show ipv6 mld-snooping group ff0e::ef00:a096 tracking
Display group ff0e::ef00:a096 in all interfaces with tracking enabled.
p-:physical, ST:static, QR:querier, EX:exclude, IN:include, Y:yes, N:no
VL1 : 1 grp, 1 grp-port, tracking_enabled
      group                                 p-port ST QR life mode source
1     ff0e::ef00:a096                       1/7    N  Y  80   EX   0
    receive reports from 1 clients: (age)
      (fe80::1011:1213:1415 60)
```

Syntax: show ipv6 mld-snooping group [<group-address> [detail] [tracking]]

To receive a report for a specific multicast group, enter that group's address for <group-address>.

Enter **detail** to display the source list of a specific VLAN.

Enter **tracking** for information on interfaces that are tracking-enabled.

The following table describes the information displayed by the **show ipv6 mld-snooping group** command.

This Field...	Displays...
group	The address of the IPv6 group (destination IPv6 address).
p-port	The physical port on which the group membership was received.
ST	**Yes** indicates that the MLD group was configured as a static group; **No** means it was learned from reports.

This Field...	Displays...
QR	**Yes** means the port is a querier port; **No** means it is not. A port becomes a non-querier port when it receives a query from a source with a lower source IP address than the port.
life	The number of seconds the group can remain in EXCLUDE mode. An EXCLUDE mode changes to INCLUDE if it does not receive an IS_EX or TO_EX message during a specified period of time. The default is 140 seconds. There is no `life` displayed in INCLUDE mode.
mode	The current mode of the interface: INCLUDE or EXCLUDE. If the interface is in INCLUDE mode, it admits traffic only from the source list. If the interface is in EXCLUDE mode, it denies traffic from the source list and accepts the rest.
source	Identifies the source list that will be included or excluded on the interface. An MLDv1 group is in EXCLUDE mode with a source of 0. The group excludes traffic from 0 (zero) source list, which actually means that all traffic sources are included.
group	If you requested a *detailed* report, the following information is displayed: • The multicast group address • The mode of the group • Sources from which traffic will be admitted (INCLUDE) or denied (EXCLUDE) on the interface. • The life of each source list. If you requested a *tracking/fast leave* report, the clients from which reports were received are identified.

Displaying MLD Snooping Mcache Information

The MLD snooping mcache contains multicast forwarding information for VLANs. To display information in the multicast forwarding mcache, enter the following command:

```
FastIron#show ipv6 mld-snooping mcache
Example: (S G) cnt=: (S G) are the lowest 32 bits, cnt: SW proc. count
    OIF: 1/22 TR(1/32,1/33), TR is trunk, 1/32 primary, 1/33 output
vlan 1, has 2 cache
1    (abcd:ef50 0:100), cnt=121
     OIF: 1/11 1/9
     age=0s up-time=120s vidx=4130 (ref-cnt=1)
2    (abcd:ef50 0:101), cnt=0
     OIF: entire vlan
     age=0s up-time=0s vidx=8191 (ref-cnt=1)
vlan 70, has 0 cache
```

Syntax: show ipv6 mld-snooping mcache

The following table describes the output from the **ipv6 mld-snooping mcache** command.

This Field...	DIsplays...
(abcd:ef50 0:100):	The lowest 32 bits of source and group. It is displayed in XXXX:XXXX hex format. Here XXXX is a 16-bit hex number.
cnt	The number of packets processed in software.

This Field...	DIsplays...
OIF	Output interfaces. *Entire vlan* means that static groups apply to the entire VLAN.
age	The mcache age in seconds. The mcache is reset to 0 if traffic continues to arrive, otherwise it is aged out when it reaches the time defined by **ipv6 mld-snooping mcache-age**.
uptime	The up time of this mcache in seconds.
vidx	The vidx is shared among mcaches using the same output interfaces. The vidx specifies the output port list, which shows the index. Valid range is from 4096 to 8191.
ref-cnt	The number of mcaches using this vidx.

Displaying Software Resource Usage for VLANs

To display information about the software resources used, enter the following command:

```
FastIron#show ipv6 mld-snooping resource
                      alloc in-use  avail get-fail    limit  get-mem  size init
mld group               512      9    503        0    32000      272    28  256
mld phy port           1024     16   1008        0   200000      279    21 1024
snoop group hash        512      9    503        0    59392      272    20  256
…. Entries deleted
total pool memory 194432 bytes
has total 1 forwarding hash
Available vidx: 4061
```

Syntax: show ipv6 mld-snooping resource

The following table describes the output from the show **ipv6 mld-snooping resource** command:

This Field...	Displays...
alloc	The allocated number of units.
in-use	The number of units which are currently used.
avail	The number of available units.
get-fail	The number of resource failures. **NOTE:** It is important to pay close attention to this field.
limit	The upper limit of this expandable field. The MLD group limit is configured using the **system-max mld-max-group-addr** command. The snoop mcache entry limit is configured using the **system-max mld-snoop-mcache** command.
get-mem	The current memory allocation. This number should continue to increase.
size	The size of a unit (in bytes).
init	The initial allocated amount of memory. **NOTE:** This number can be increased. (More memory can be allocated if necessary.)

This Field...	Displays...
Available vidx	The output interface (OIF) port mask used by mcache. The entire device has a maximum of 4096 vidx. Different mcaches with the same OIF share the same vidx. If vidx is not available, the stream cannot be hardware-switched.

Displaying Status of MLD Snooping Traffic

To display status information for MLD snooping traffic, enter the following command:

```
FastIron#show ipv6 mld-snooping traffic

MLD snooping: Total Recv: 32208, Xmit: 166
Q: query, Qry: general Q,  G-Qry: group Q,  GSQry: group-source Q, Mbr: member
Recv      QryV1     QryV2     G-Qry     GSQry     MbrV1     MbrV2     Leave
VL1           0         0         0         0     31744       208       256
VL70          0         0         0         0         0         0         0
Recv       IsIN      IsEX      ToIN      ToEX     ALLOW     BLOCK   Pkt-Err
VL1        1473     31784         0         1         1         7         0
VL70          0         0         0         0         0         0         0
Send      QryV1     QryV2     G-Qry     GSQry     MbrV1     MbrV2
VL1           0         0       166         0         0         0
VL70          0         0         0         0         0         0
```

Syntax: show ipv6 mld-snooping traffic

The following table describes the information displayed by the **show ipv6 mld-snooping traffic** command.

This Field	Displays
Q	Query
Qry	General Query
QryV1	Number of general MLDv1 queries received or sent.
QryV2	Number of general MLDv2 snooping queries received or sent.
G-Qry	Number of group specific queries received or sent.
GSQry	Number of group source specific queries received or sent.
MBR	The membership report.
MbrV1	The MLDv1 membership report.
MbrV2	The MLDv2 membership report.
IsIN	Number of source addresses that were included in the traffic.
IsEX	Number of source addresses that were excluded in the traffic.
ToIN	Number of times the interface mode changed from EXCLUDE to INCLUDE.
ToEX	Number of times the interface mode changed from INCLUDE to EXCLUDE.
ALLO	Number of times additional source addresses were allowed on the interface.
BLK	Number of times sources were removed from an interface.

This Field	Displays
Pkt-Err	Number of packets having errors such as checksum errors.

Displaying MLD Snooping Information by VLAN

You can display MLD snooping information for all VLANs or for a specific VLAN. For example, to display MLD snooping information for VLAN 70, enter the following command:

```
FastIron#show ipv6 mld-snooping vlan 70
version=1, query-t=60, group-aging-t=140, max-resp-t=3, other-qr-present-t=123
VL70: cfg V2, vlan cfg passive, 2 grp, 0 (SG) cache, rtr ports,
    router ports: 1/36(120) fe80::2e0:52ff:fe00:9900,
  1/26 has 2 grp, non-QR (passive), cfg V1
  1/26 has 2 grp, non-QR (passive), cfg V1
    group: ff10:1234::5679, life = 100
    group: ff10:1234::5678, life = 100
  1/35 has 0 grp, non-QR (QR=fe80::2e0:52ff:fe00:9900, age=20), dft V2 trunk
```

Syntax: show ipv6 mld-snooping vlan [<vlan-id>]

If you do not specify a **vlan-id**, information for all VLANs is displayed.

The following table describes information displayed by the **show ipv6 mld-snooping vlan** command.

This Field	Displays
version	The MLD version number.
query-t	How often a querier sends a general query on the interface.
group-aging-t	Number of seconds membership groups can be members of this group before aging out.
rtr-port	The router ports which are the ports receiving queries. The display `router ports: 1/36(120) fe80::2e0:52ff:fe00:9900` means port 1/36 has a querier with fe80::2e0:52ff:fe00:9900 as the link-local address, and the remaining life is 120 seconds.
max-resp-t	The maximum number of seconds a client can wait before it replies to the query.
non-QR	Indicates that the port is a non-querier.
QR	Indicates that the port is a querier.

Clearing MLD Snooping Counters and Mcache

The clear commands for MLD snooping should only be used in troubleshooting situations or when recovering from error conditions.

Clearing MLD Counters on all VLANs

To clear MLD snooping error and traffic counters on all VLANs, enter the following command:

```
FastIron#clear ipv6 mld-snooping counters
```

Syntax: clear ipv6 mld-snooping counters

Clearing the mcache on all VLANs

To clear the mcache on all VLANs, enter the following command:

```
FastIron#clear ipv6 mld-snooping mcache
```

Syntax: clear ipv6 mld-snooping mcache

Clearing the mcache on a Specific VLAN

To clear the mcache on a specific VLAN, enter the following command:

```
FastIron#clear ipv6 mld-snooping vlan 10 mcache
```

Syntax: clear ipv6 mld-snooping vlan <vlan-id> mcache

The <vlan-id> parameter specifies the specific VLAN from which to clear the cache.

Clearing Traffic Counters on a Specific VLAN

To clear the traffic counters on a specific VLAN, enter the following command:

```
FastIron#clear ipv6 mld-snooping vlan 10 traffic
```

Syntax: clear ipv6 mld-snooping vlan <vlan-id> traffic

The <vlan-id> parameter specifies the specific VLAN from which to clear the traffic counters.

Chapter 33
Configuring RIP

This chapter describes how to configure RIP on a Foundry Layer 3 Switch.

Platform Support:

- FESX/FSX/FWSX devices running software release 02.0.00 and later

RIP Overview

Routing Information Protocol (RIP) is an IP route exchange protocol that uses a ***distance vector*** (a number representing distance) to measure the cost of a given route. The ***cost*** is a distance vector because the cost often is equivalent to the number of router hops between the Foundry Layer 3 Switch and the destination network.

A Foundry Layer 3 Switch can receive multiple paths to a destination. The software evaluates the paths, selects the best path, and saves the path in the IP route table as the route to the destination. Typically, the best path is the path with the fewest hops. A hop is another router through which packets must travel to reach the destination. If the Foundry Layer 3 Switch receives a RIP update from another router that contains a path with fewer hops than the path stored in the Foundry Layer 3 Switch's route table, the Layer 3 Switch replaces the older route with the newer one. The Layer 3 Switch then includes the new path in the updates it sends to other RIP routers, including Foundry Layer 3 Switches.

RIP routers, including the Foundry Layer 3 Switch, also can modify a route's cost, generally by adding to it, to bias the selection of a route for a given destination. In this case, the actual number of router hops may be the same, but the route has an administratively higher cost and is thus less likely to be used than other, lower-cost routes.

A RIP route can have a maximum cost of 15. Any destination with a higher cost is considered unreachable. Although limiting to larger networks, the low maximum hop count prevents endless loops in the network.

Foundry Layer 3 Switches support the following RIP versions:

- Version 1
- V1 compatible with V2
- Version 2 (the default)

ICMP Host Unreachable Message for Undeliverable ARPs

If the router receives an ARP request packet that it is unable to deliver to the final destination because of the ARP timeout and no ARP response is received (router knows of no route to the destination address), the router sends an ICMP Host Unreachable message to the source.

RIP Parameters and Defaults

The following tables list the RIP parameters, their default values, and where to find configuration information.

RIP Global Parameters

Table 33.1 lists the global RIP parameters and their default values, and indicates where you can find configuration information.

Table 33.1: RIP Global Parameters

Parameter	Description	Default	See page...
RIP state	The global state of the protocol **Note**: You also must enable the protocol on individual interfaces. Globally enabling the protocol does not allow interfaces to send and receive RIP information. See Table 33.2 on page 33-3.	Disabled	33-3
Administrative distance	The administrative distance is a numeric value assigned to each type of route on the router. When the router is selecting from among multiple routes (sometimes of different origins) to the same destination, the router compares the administrative distances of the routes and selects the route with the lowest administrative distance. This parameter applies to routes originated by RIP. The administrative distance stays with a route when it is redistributed into other routing protocols.	120	33-5
Redistribution	RIP can redistribute routes from other routing protocols such as OSPF and BGP4 into RIP. A redistributed route is one that a router learns through another protocol, then distributes into RIP.	Disabled	33-5
Redistribution metric	RIP assigns a RIP metric (cost) to each external route redistributed from another routing protocol into RIP. An external route is a route with at least one hop (packets must travel through at least one other router to reach the destination). This parameter applies to routes that are redistributed from other protocols into RIP.	1 (one)	33-6
Update interval	How often the router sends route updates to its RIP neighbors	30 seconds	33-7
Learning default routes	The router can learn default routes from its RIP neighbors. **Note**: You also can enable or disable this parameter on an individual interface basis. See Table 33.2 on page 33-3.	Disabled	33-7
Advertising and learning with specific neighbors	The Layer 3 Switch learns and advertises RIP routes with all its neighbors by default. You can prevent the Layer 3 Switch from advertising routes to specific neighbors or learning routes from specific neighbors.	Learning and advertising permitted for all neighbors	33-8

RIP Interface Parameters

Table 33.2 lists the interface-level RIP parameters and their default values, and indicates where you can find configuration information.

.

Table 33.2: RIP Interface Parameters

Parameter	Description	Default	See page...
RIP state and version	The state of the protocol and the version that is supported on the interface. The version can be one of the following: • Version 1 only • Version 2 only • Version 1, but also compatible with version 2 **Note**: You also must enable RIP globally.	Disabled	33-3
Metric	A numeric cost the router adds to RIP routes learned on the interface. This parameter applies only to RIP routes.	1 (one)	33-4
Learning default routes	Locally overrides the global setting. See Table 33.1 on page 33-2.	Disabled	33-7
Loop prevention	The method a router uses to prevent routing loops caused by advertising a route on the same interface as the one on which the router learned the route. • Split horizon – The router does not advertise a route on the same interface as the one on which the router learned the route. • Poison reverse – The router assigns a cost of 16 (“infinite” or “unreachable”) to a route before advertising it on the same interface as the one on which the router learned the route.	Poison reverse **Note**: Enabling split horizon disables poison reverse on the interface.	33-8
Advertising and learning specific routes	You can control the routes that a Layer 3 Switch learns or advertises.	The Layer 3 Switch learns and advertises all RIP routes on all interfaces.	33-9

Configuring RIP Parameters

Use the following procedures to configure RIP parameters on a system-wide and individual interface basis.

Enabling RIP

RIP is disabled by default. To enable it, use the following method.

NOTE: You must enable the protocol globally and also on individual interfaces on which you want to advertise RIP. Globally enabling the protocol does not enable it on individual interfaces.

To enable RIP globally, enter the following command:

```
FastIron(config)#router rip
```

Syntax: [no] router rip

After globally enabling the protocol, you must enable it on individual interfaces. You can enable the protocol on physical interfaces as well as virtual routing interfaces. To enable RIP on an interface, enter commands such as the following:

```
FastIron(config)#interface ethernet 1/1
FastIron(config-if-1/1)#ip rip v1-only
```

Syntax: [no] ip rip v1-only | v1-compatible-v2 | v2-only

NOTE: You must specify the RIP version.

Configuring Metric Parameters

By default, a Foundry Layer 3 Switch port increases the cost of a RIP route that is learned on the port by one. You can configure individual ports to add more than one to a learned route's cost. In addition, you can configure a RIP offset list to increase the metric for learned or advertised routes based on network address.

Changing the Cost of Routes Learned on a Port

By default, a Foundry Layer 3 Switch port increases the cost of a RIP route that is learned on the port. The Layer 3 Switch increases the cost by adding one to the route's metric before storing the route.

You can change the amount that an individual port adds to the metric of RIP routes learned on the port. To do so, use the following method.

NOTE: RIP considers a route with a metric of 16 to be unreachable. Use this metric only if you do not want the route to be used. In fact, you can prevent the Layer 3 Switch from using a specific port for routes learned though that port by setting its metric to 16.

To increase the cost a port adds to RIP routes learned in that port, enter commands such as the following:

```
FastIron(config)#interface ethernet 6/1
FastIron(config-if-6/1)#ip metric 5
```

These commands configure port 6/1 to add 5 to the cost of each route learned on the port.

Syntax: ip metric <1-16>

Configuring a RIP Offset List

A RIP offset list allows you to add to the metric of specific inbound or outbound routes learned or advertised by RIP. RIP offset lists provide a simple method for adding to the cost of specific routes and therefore biasing the Layer 3 Switch's route selection away from those routes.

An offset list consists of the following parameters:

- An ACL that specifies the routes to which to add the metric.
- The direction:
  - In applies to routes the Layer 3 Switch learns from RIP neighbors.
  - Out applies to routes the Layer 3 Switch is advertising to its RIP neighbors.
- The type and number of a specific port to which the offset list applies (optional).

The software adds the offset value to the routing metric (cost) of the routes that match the ACL. If a route matches both a global offset list and an interface-based offset list, the interface-based offset list takes precedence. The interface-based offset list's metric is added to the route in this case.

You can configure up to 24 global RIP offset lists and up to 24 RIP offset lists on each interface.

To configure a global RIP offset list, enter commands such as the following:

```
FastIron(config)#access-list 21 deny 160.1.0.0 0.0.255.255
FastIron(config)#access-list 21 permit any
FastIron(config)#router rip
FastIron(config-rip-router)#offset-list 21 out 10
```

The commands in this example configure a standard ACL. The ACL matches on all IP networks except 160.1.x.x. When the Layer 3 Switch advertises a route that matches ACL 21, the offset list adds 10 to the route's metric.

Syntax: [no] <acl-number-or-name> in | out offset [ethernet [<slotnum>/]<portnum>]

In the following example, the Layer 3 Switch uses ACL 21 to add 10 to the metric of routes received on Ethernet port 2/1.

```
FastIron(config-rip-router)#offset-list 21 in ethernet 2/1
```

Changing the Administrative Distance

By default, the Layer 3 Switch assigns the default RIP administrative distance (120) to RIP routes. When comparing routes based on administrative distance, the Layer 3 Switch selects the route with the lower distance. You can change the administrative distance for RIP routes.

NOTE: See "Changing Administrative Distances" on page 38-28 for the default distances for all route sources.

To change the administrative distance for RIP routes, enter a command such as the following:

```
FastIron(config-rip-router)#distance 140
```

This command changes the administrative distance to 140 for all RIP routes.

Syntax: [no] distance <num>

Configuring Redistribution

You can configure the Layer 3 Switch to redistribute routes learned through Open Shortest Path First (OSPF) or Border Gateway Protocol version 4 (BGP4) into RIP. When you redistribute a route from one of these other protocols into RIP, the Layer 3 Switch can use RIP to advertise the route to its RIP neighbors.

To configure redistribution, perform the following tasks:

- Configure redistribution filters (optional). You can configure filters to permit or deny redistribution for a route based on its origin (OSPF, BGP4, and so on), the destination network address, and the route's metric. You also can configure a filter to set the metric based on these criteria.
- Change the default redistribution metric (optional). The Layer 3 Switch assigns a RIP metric of one to each redistributed route by default. You can change the default metric to a value up to 16.
- Enable redistribution.

  NOTE: Do not enable redistribution until you configure the other redistribution parameters.

Configuring Redistribution Filters

RIP redistribution filters apply to all interfaces. The software uses the filters in ascending numerical order and immediately takes the action specified by the filter. Thus, if filter 1 denies redistribution of a given route, the software does not redistribute the route, regardless of whether a filter with a higher ID would permit redistribution of that route.

NOTE: The default redistribution action is permit, even after you configure and apply redistribution filters to the virtual routing interface. If you want to tightly control redistribution, apply a filter to deny all routes as the last filter (the filter with the highest ID), then apply filters with lower filter IDs to allow specific routes.

To configure a redistribution filter, enter a command such as the following:

```
FastIron(config-rip-router)#deny redistribute 2 all address 207.92.0.0 255.255.0.0
```

This command denies redistribution for all types of routes to the 207.92.x.x network.

Syntax: [no] permit | deny redistribute <filter-num> all | bgp | ospf | static address <ip-addr> <ip-mask> [match-metric <value> | set-metric <value>]

The <filter-num> specifies the redistribution filter ID. The software uses the filters in ascending numerical order. Thus, if filter 1 denies a route from being redistributed, the software does not redistribute that route even if a filter with a higher ID permits redistribution of the route.

The **all** parameter applies redistribution to all route types.

The **bgp** parameter applies redistribution to BGP4 routes only.

The **ospf** parameter applies redistribution to OSPF routes only.

The **static** parameter applies redistribution to IP static routes only.

The **address** <ip-addr> <ip-mask> parameters apply redistribution to the specified network and subnet address. Use 0 to specify “any”. For example, “207.92.0.0 255.255.0.0“ means “any 207.92.x.x subnet”. However, to specify any subnet (all subnets match the filter), enter “address 255.255.255.255 255.255.255.255”.

The **match-metric** <value> parameter applies the redistribution filter only to those routes with the specified metric value; possible values are from 1 – 15.

The **set-metric** <value> parameter sets the RIP metric value that will be applied to those routes imported into RIP.

The following command denies redistribution into RIP for all OSPF routes:

```
FastIron(config-rip-router)#deny redistribute 3 ospf address 207.92.0.0 255.255.0.0
```

The following command denies redistribution for all OSPF routes that have a metric of 10:

```
FastIron(config-rip-router)#deny redistribute 3 ospf address 207.92.0.0 255.255.0.0
match-metric 10
```

The following commands deny redistribution of all routes except routes for 10.10.10.x and 20.20.20.x:

```
FastIron(config-rip-router)#deny redistribute 64 static address 255.255.255.255 255.255.255.255
FastIron(config-rip-router)#permit redistribute 1 static address 10.10.10.0 255.255.255.0
FastIron(config-rip-router)#permit redistribute 2 static address 20.20.20.0 255.255.255.0
```

NOTE: This example assumes that the highest RIP redistribution filter ID configured on the device is 64.

Changing the Redistribution Metric

When the Layer 3 Switch redistributes a route into RIP, the software assigns a RIP metric (cost) to the route. By default, the software assigns a metric of one to each route that is redistributed into RIP. You can increase the metric that the Layer 3 Switch assigns, up to 15.

To change the RIP metric the Layer 3 Switch assigns to redistributed routes, enter a command such as the following:

```
FastIron(config-rip-router)#default-metric 10
```

This command assigns a RIP metric of 10 to each route that is redistributed into RIP.

Syntax: [no] default-metric <1-15>

Enabling Redistribution

After you configure redistribution parameters, you need to enable redistribution.

To enable RIP redistribution, enter the following command:

```
FastIron(config-rip-router)#redistribution
```

Syntax: [no] redistribution

Removing a RIP Redistribution Deny Filter

To remove a previously configured RIP redistribution deny filter, do the following:

1. Remove the RIP redistribution deny filter.
2. Disable the redistribution function.
3. Re-enable redistribution.

The following shows an example of how to remove a RIP redistribution deny filter:

```
FastIron(config-rip-router)#no deny redistribute 2 all address 207.92.0.0 255.255.0.0
FastIron(config-rip-router)#no redistribution
FastIron(config-rip-router)#redistribution
```

Configuring Route Learning and Advertising Parameters

By default, a Foundry Layer 3 Switch learns routes from all its RIP neighbors and advertises RIP routes to those neighbors.

You can configure the following learning and advertising parameters:

- Update interval – The update interval specifies how often the Layer 3 Switch sends RIP route advertisements to its neighbors. The default is 30 seconds. You can change the interval to a value from 1 – 1000 seconds.
- Learning and advertising of RIP default routes – The Layer 3 Switch learns and advertises RIP default routes by default. You can disable learning and advertising of default routes on a global or individual interface basis.
- Learning of standard RIP routes – By default, the Layer 3 Switch can learn RIP routes from all its RIP neighbors. You can configure RIP neighbor filters to explicitly permit or deny learning from specific neighbors.

Changing the Update Interval for Route Advertisements

The update interval specifies how often the Layer 3 Switch sends route advertisements to its RIP neighbors. You can specify an interval from 1 – 1000 seconds. The default is 30 seconds.

To change the RIP update interval, enter a command such as the following:

```
FastIron(config-rip-router)#update 120
```

This command configures the Layer 3 Switch to send RIP updates every 120 seconds.

Syntax: update-time <1-1000>

Enabling Learning of RIP Default Routes

By default, the Layer 3 Switch does not learn RIP default routes. You can enable learning of RIP default routes on a global or interface basis.

To enable learning of default RIP routes on a global basis, enter the following command:

```
FastIron(config-rip-router)#learn-default
```

Syntax: [no] learn-default

To enable learning of default RIP routes on an interface basis, enter commands such as the following:

```
FastIron(config)#interface ethernet 1
FastIron(config-if-1)#ip rip learn-default
```

Syntax: [no] ip rip learn-default

Configuring a RIP Neighbor Filter

By default, a Foundry Layer 3 Switch learns RIP routes from all its RIP neighbors. Neighbor filters allow you to specify the neighbor routers from which the Foundry device can receive RIP routes. Neighbor filters apply globally to all ports.

To configure a RIP neighbor filters, enter a command such as the following:

```
FastIron(config-rip-router)#neighbor 1 deny any
```

Syntax: [no] neighbor <filter-num> permit | deny <source-ip-address> | any

This command configures the Layer 3 Switch so that the device does not learn any RIP routes from any RIP neighbors.

The following commands configure the Layer 3 Switch to learn routes from all neighbors except 192.168.1.170. Once you define a RIP neighbor filter, the default action changes from learning all routes from all neighbors to denying all routes from all neighbors except the ones you explicitly permit. To deny learning from a specific neighbor but allow all other neighbors, you must add a filter that allows learning from all neighbors. Be sure to add the filter to permit all neighbors last (the one with the highest filter number). Otherwise, the software can match on the permit all filter instead of a filter that denies a specific neighbor, and learn routes from that neighbor.

```
FastIron(config-rip-router)#neighbor 2 deny 192.16.1.170
FastIron(config-rip-router)#neighbor 1024 permit any
```

Changing the Route Loop Prevention Method

RIP can use the following methods to prevent routing loops:

- Split horizon – The Layer 3 Switch does not advertise a route on the same interface as the one on which the router learned the route.
- Poison reverse – The Layer 3 Switch assigns a cost of 16 (“infinite” or “unreachable”) to a route before advertising it on the same interface as the one on which the router learned the route. This is the default.

These loop prevention methods are configurable on an individual interface basis. One of the methods is always in effect on an interface enabled for RIP. If you disable one method, the other method is enabled.

NOTE: These methods may be used in addition to RIP’s maximum valid route cost of 15.

To disable poison reverse and enable split horizon on an interface, enter commands such as the following:

```
FastIron(config)#interface ethernet 1/1
FastIron(config-if-1/1)#no ip rip poison-reverse
```

Syntax: [no] ip rip poison-reverse

To disable split horizon and enable poison reverse on an interface, enter commands such as the following:

```
FastIron(config)#interface ethernet 1/1
FastIron(config-if-1/1)#ip rip poison-reverse
```

Suppressing RIP Route Advertisement on a VRRP or VRRPE Backup Interface

NOTE: This section applies only if you configure the Layer 3 Switch for Virtual Router Redundancy Protocol (VRRP) or VRRP Extended (VRRPE). See “Configuring VRRP and VRRPE” on page 37-1.

Normally, a VRRP or VRRPE Backup includes route information for the virtual IP address (the backed up interface) in RIP advertisements. As a result, other routers receive multiple paths for the backed up interface and might sometimes unsuccessfully use the path to the Backup rather than the path to the Master.

You can prevent the Backups from advertising route information for the backed up interface by enabling suppression of the advertisements.

To suppress RIP advertisements for the backed up interface, enter the following commands:

```
FastIron(config)#router rip
FastIron(config-rip-router)#use-vrrp-path
```

Syntax: [no] use-vrrp-path

The syntax is the same for VRRP and VRRPE.

Configuring RIP Route Filters

You can configure RIP route filters to permit or deny learning or advertising of specific routes. Configure the filters globally, then apply them to individual interfaces. When you apply a RIP route filter to an interface, you specify whether the filter applies to learned routes (in) or advertised routes (out).

NOTE: A route is defined by the destination's IP address and network mask.

NOTE: By default, routes that do not match a route filter are learned or advertised. To prevent a route from being learned or advertised, you must configure a filter to deny the route.

To configure RIP filters, enter commands such as the following:

```
FastIron(config-rip-router)#filter 1 permit 192.53.4.1 255.255.255.0
FastIron(config-rip-router)#filter 2 permit 192.53.5.1 255.255.255.0
FastIron(config-rip-router)#filter 3 permit 192.53.6.1 255.255.255.0
FastIron(config-rip-router)#filter 4 deny 192.53.7.1 255.255.255.0
```

These commands explicitly permit RIP routes to three networks, and deny the route to one network.

Since the default action is permit, all other routes (routes not explicitly permitted or denied by the filters) can be learned or advertised.

Syntax: filter <filter-num> permit | deny <source-ip-address> | any <source-mask> | any [log]

Applying a RIP Route Filter to an Interface

Once you define RIP route filters, you must assign them to individual interfaces. The filters do not take effect until you apply them to interfaces. When you apply a RIP route filter, you also specify whether the filter applies to learned routes or advertised routes:

- Out filters apply to routes the Layer 3 Switch advertises to its neighbor on the interface.
- In filters apply to routes the Layer 3 Switch learns from its neighbor on the interface.

To apply RIP route filters to an interface, enter commands such as the following:

```
FastIron(config)#interface ethernet 1/2
FastIron(config-if-1/2)#ip rip filter-group in 2 3 4
```

Syntax: [no] ip rip filter-group in | out <filter-list>

These commands apply RIP route filters 2, 3, and 4 to all routes learned from the RIP neighbor on port 1/2.

Displaying RIP Filters

To display the RIP filters configured on the router, enter the following command at any CLI level:

```
FastIron#show ip rip

                RIP Route Filter Table
  Index   Action   Route IP Address    Subnet Mask
  1       deny     any                 any

                RIP Neighbor Filter Table
  Index   Action   Neighbor IP Address
  1       permit   any
```

Syntax: show ip rip

This display shows the following information.

Table 33.3: CLI Display of RIP Filter Information

This Field...	Displays...
Route filters The rows underneath "RIP Route Filter Table" list the RIP route filters. If no RIP route filters are configured on the device, the following message is displayed instead: "No Filters are configured in RIP Route Filter Table".	
Index	The filter number. You assign this number when you configure the filter.
Action	The action the router takes if a RIP route packet matches the IP address and subnet mask of the filter. The action can be one of the following: • deny – RIP route packets that match the address and network mask information in the filter are dropped. If applied to an interface's outbound filter group, the filter prevents the router from advertising the route on that interface. If applied to an interface's inbound filter group, the filter prevents the router from adding the route to its IP route table. • permit – RIP route packets that match the address and network mask information are accepted. If applied to an interface's outbound filter group, the filter allows the router to advertise the route on that interface. If applied to an interface's inbound filter group, the filter allows the router to add the route to its IP route table.
Route IP Address	The IP address of the route's destination network or host.
Subnet Mask	The network mask for the IP address.
Neighbor filters The rows underneath "RIP Neighbor Filter Table" list the RIP neighbor filters. If no RIP neighbor filters are configured on the device, the following message is displayed instead: "No Filters are configured in RIP Neighbor Filter Table".	
Index	The filter number. You assign this number when you configure the filter.

Table 33.3: CLI Display of RIP Filter Information (Continued)

This Field...	Displays...
Action	The action the router takes for RIP route packets to or from the specified neighbor: • deny – If the filter is applied to an interface's outbound filter group, the filter prevents the router from advertising RIP routes to the specified neighbor on that interface. If the filter is applied to an interface's inbound filter group, the filter prevents the router from receiving RIP updates from the specified neighbor. • permit – If the filter is applied to an interface's outbound filter group, the filter allows the router to advertise RIP routes to the specified neighbor on that interface. If the filter is applied to an interface's inbound filter group, the filter allows the router to receive RIP updates from the specified neighbor.
Neighbor IP Address	The IP address of the RIP neighbor.

Displaying CPU Utilization Statistics

You can display CPU utilization statistics for RIP and other IP protocols.

To display CPU utilization statistics for RIP for the previous one-second, one-minute, five-minute, and fifteen-minute intervals, enter the following command at any level of the CLI:

```
FastIron#show process cpu
Process Name   5Sec(%)  1Min(%)  5Min(%)  15Min(%)  Runtime(ms)
ARP              0.01     0.03     0.09     0.22            9
BGP              0.04     0.06     0.08     0.14            13
GVRP             0.00     0.00     0.00     0.00            0
ICMP             0.00     0.00     0.00     0.00            0
IP               0.00     0.00     0.00     0.00            0
OSPF             0.00     0.00     0.00     0.00            0
RIP              0.04     0.07     0.08     0.09            7
STP              0.00     0.00     0.00     0.00            0
VRRP             0.00     0.00     0.00     0.00            0
```

If the software has been running less than 15 minutes (the maximum interval for utilization statistics), the command indicates how long the software has been running. Here is an example:

```
FastIron#show process cpu
The system has only been up for 6 seconds.
Process Name   5Sec(%)  1Min(%)  5Min(%)  15Min(%)  Runtime(ms)
ARP              0.01     0.00     0.00     0.00            0
BGP              0.00     0.00     0.00     0.00            0
GVRP             0.00     0.00     0.00     0.00            0
ICMP             0.01     0.00     0.00     0.00            1
IP               0.00     0.00     0.00     0.00            0
OSPF             0.00     0.00     0.00     0.00            0
RIP              0.00     0.00     0.00     0.00            0
STP              0.00     0.00     0.00     0.00            0
VRRP             0.00     0.00     0.00     0.00            0
```

To display utilization statistics for a specific number of seconds, enter a command such as the following:

```
FastIron#show process cpu 2
Statistics for last 1 sec and 80 ms
Process Name   Sec(%)   Time(ms)
ARP              0.00        0
BGP              0.00        0
GVRP             0.00        0
ICMP             0.01        1
IP               0.00        0
OSPF             0.00        0
RIP              0.00        0
STP              0.01        0
VRRP             0.00        0
```

When you specify how many seconds' worth of statistics you want to display, the software selects the sample that most closely matches the number of seconds you specified. In this example, statistics are requested for the previous two seconds. The closest sample available is actually for the previous 1 second plus 80 milliseconds.

Syntax: show process cpu [<num>]

The <num> parameter specifies the number of seconds and can be from 1 – 900. If you use this parameter, the command lists the usage statistics only for the specified number of seconds. If you do not use this parameter, the command lists the usage statistics for the previous one-second, one-minute, five-minute, and fifteen-minute intervals.

Chapter 34
Configuring RIPng

This chapter describes how to configure RIPng on a Foundry IPv6 Layer 3 Switch.

Platform Support:

- FESX and FSX IPv6 devices running software release 04.1.00 and later – L3

RIPng Overview

Routing Information Protocol (RIP) is an IP route exchange protocol that uses a distance vector (a number representing a distance) to measure the cost of a given route. RIP uses a hop count as its cost or metric.

IPv6 RIP, known as ***Routing Information Protocol Next Generation*** or ***RIPng*** functions similarly to IPv4 RIP Version 2. RIPng supports IPv6 addresses and prefixes and introduces some new commands that are specific to RIPng. This chapter describes the commands that are specific to RIPng. This section does not describe commands that apply to both IPv4 RIP and RIPng.

RIPng maintains a ***Routing Information Database (RIB)***, which is a local route table. The local RIB contains the lowest-cost IPv6 routes learned from other RIP routers. In turn, RIPng attempts to add routes from its local RIB into the main IPv6 route table.

This section describes the following:

- How to configure RIPng
- How to clear RIPng information from the RIPng route table
- How to display RIPng information and statistics

Configuring RIPng

To configure RIPng, you must enable RIPng globally on the Foundry device and on individual router interfaces. The following configuration tasks are optional:

- Change the default settings of RIPng timers.
- Configure how the Foundry device learns and advertises routes.
- Configure which routes are redistributed into RIPng from other sources.
- Configure how the Foundry device distributes routes via RIPng.
- Configure poison reverse parameters.

Enabling RIPng

Before configuring the Foundry device to run RIPng, you must do the following:

- Enable the forwarding of IPv6 traffic on the Foundry device using the **ipv6 unicast-routing** command.
- Enable IPv6 on each interface over which you plan to enable RIPng. You enable IPv6 on an interface by configuring an IPv6 address or explicitly enabling IPv6 on that interface.

For more information about performing these configuration tasks, see the chapter *IPv6 Management on FastIron Devices* in the *Foundry FastIron Configuration Guide*.

By default, RIPng is disabled. To enable RIPng, you must enable it globally on the Foundry device and also on individual router interfaces.

NOTE: You are required to configure a router ID when running only IPv6 routing protocols.

NOTE: Enabling RIPng globally on the Foundry device does not enable it on individual router interfaces.

To enable RIPng globally, enter the following command:

```
FastIron(config-rip-router)#ipv6 router rip
FastIron(config-ripng-router)#
```

After you enter this command, the Foundry device enters the RIPng configuration level, where you can access several commands that allow you to configure RIPng.

Syntax: [no] ipv6 router rip

To disable RIPng globally, use the **no** form of this command.

After enabling RIPng globally, you must enable it on individual router interfaces. You can enable it on physical as well as virtual routing interfaces. For example, to enable RIPng on Ethernet interface 3/1, enter the following commands:

```
FastIron(config)# interface ethernet 3/1
FastIron(config-if-e100-3/1)# ipv6 rip enable
```

Syntax: [no] ipv6 rip enable

To disable RIPng on an individual router interface, use the **no** form of this command.

Configuring RIPng Timers

Table 34.1 describes the RIPng timers and provides their defaults.

Table 34.1:RIPng timers

Timer	Description	Default
Update	Amount of time (in seconds) between RIPng routing updates.	30 seconds.
Timeout	Amount of time (in seconds) after which a route is considered unreachable.	180 seconds.
Hold-down	Amount of time (in seconds) during which information about other paths is ignored.	180 seconds.
Garbage-collection	Amount of time (in seconds) after which a route is removed from the routing table.	120 seconds.

You can adjust these timers for RIPng. Before doing so, keep the following caveats in mind:

- If you adjust these RIPng timers, Foundry strongly recommends setting the same timer values for all routers and access servers in the network.
- Setting the update timer to a shorter interval can cause the routers to spend excessive time updating the IPv6 route table.
- Foundry recommends setting the timeout timer value to at least three times the value of the update timer.
- Foundry recommends a shorter hold-down timer interval, because a longer interval can cause delays in RIPng convergence.

The following example sets updates to be broadcast every 45 seconds. If a route is not heard from in 135 seconds, the route is declared unusable. Further information is suppressed for an additional 10 seconds. Assuming no updates, the route is flushed from the routing table 20 seconds after the end of the hold-down period.

```
FastIron(config)# ipv6 router rip
FastIron(config-ripng-router)# timers 45 135 10 20
```

Syntax: [no] timers <update-timer> <timeout-timer> <hold-down-timer> <garbage-collection-timer>

Possible values for the timers are as follows:

- Update timer: 3 – 65535 seconds.
- Timeout timer: 9 – 65535 seconds.
- Hold-down timer: 9 – 65535 seconds.
- Garbage-collection timer: 9 – 65535 seconds.

NOTE: You must enter a value for each timer, even if you want to retain the current setting of a particular timer.

To return to the default values of the RIPng timers, use the **no** form of this command.

Configuring Route Learning and Advertising Parameters

You can configure the following learning and advertising parameters:

- Learning and advertising of RIPng default routes.
- Advertising of IPv6 address summaries.
- Metric of routes learned and advertised on a router interface.

Configuring Default Route Learning and Advertising

By default, the Foundry device does not learn IPv6 default routes (::/0). You can originate default routes into RIPng, which causes individual router interfaces to include the default routes in their updates. When configuring the origination of the default routes, you can also do the following:

- Suppress all other routes from the updates.
- Include all other routes in the updates.

For example, to originate default routes in RIPng and suppress all other routes in updates sent from Ethernet interface 3/1, enter the following commands:

```
FastIron(config)# interface ethernet 3/1
FastIron(config-if-e100-3/1)# ipv6 rip default-information only
```

To originate IPv6 default routes and include all other routes in updates sent from Ethernet interface 3/1, enter the following commands:

```
FastIron(config)# interface ethernet 3/1
FastIron(config-if-e100-3/1)# ipv6 rip default-information originate
```

Syntax: [no] ipv6 rip default-information only | originate

The **only** keyword originates the default routes and suppresses all other routes from the updates.

The **originate** keyword originates the default routes and includes all other routes in the updates.

To remove the explicit default routes from RIPng and suppress advertisement of these routes, use the **no** form of this command.

Advertising IPv6 Address Summaries

You can configure RIPng to advertise a summary of IPv6 addresses from a router interface and to specify an IPv6 prefix that summarizes the routes.

If a route's prefix length matches the value specified in the **ipv6 rip summary-address** command, RIPng advertises the prefix specified in the **ipv6 rip summary-address** command instead of the original route.

For example, to advertise the summarized prefix 2001:469e::/36 instead of the IPv6 address 2001:469e:0:adff:8935:e838:78:e0ff with a prefix length of 64 bits from Ethernet interface 3/1, enter the following commands:

```
FastIron(config)# interface ethernet 3/1
FastIron(config-if-e100-3/1)# ipv6 address 2001:469e:0:adff:8935:e838:78:
e0ff /64
FastIron(config-if-e100-3/1)# ipv6 rip summary-address 2001:469e::/36
```

Syntax: [no] ipv6 rip summary-address <ipv6-prefix>/<prefix-length>

You must specify the <ipv6-prefix> parameter in hexadecimal using 16-bit values between colons as documented in RFC 2373.

You must specify the <prefix-length> parameter as a decimal value. A slash mark (/) must follow the <ipv6-prefix> parameter and precede the <prefix-length> parameter.

To stop the advertising of the summarized IPv6 prefix, use the **no** form of this command.

Changing the Metric of Routes Learned and Advertised on an Interface

A router interface increases the metric of an incoming RIPng route it learns by an offset (the default is one). The Foundry device then places the route in the route table. When the Foundry device sends an update, it advertises the route with the metric plus the default offset of zero in an outgoing update message.

You can change the metric offset an individual interface adds to a route learned by the interface or advertised by the interface. For example, to change the metric offset for incoming routes learned by Ethernet interface 3/1 to one and the metric offset for outgoing routes advertised by the interface to three, enter the following commands:

```
FastIron(config)# interface ethernet 3/1
FastIron(config-if-e100-3/1)# ipv6 rip metric-offset 1
FastIron(config-if-e100-3/1)# ipv6 rip metric-offset out 3
```

In this example, if Ethernet interface 3/1 learns about an incoming route, it will increase the incoming metric by two (the default offset of 1 and the additional offset of 1 as specified in this example). If Ethernet interface 3/1 advertises an outgoing route, it will increase the metric by 3 as specified in this example.

Syntax: [no] ipv6 rip metric-offset [out] <1 – 16>

To return the metric offset to its default value, use the **no** form of this command.

Redistributing Routes Into RIPng

You can configure the Foundry device to redistribute routes from the following sources into RIPng:

- IPv6 static routes
- Directly connected IPv6 networks
- OSPF V3

When you redistribute a route from IPv6 or OSPF V3 into RIPng, the Foundry device can use RIPng to advertise the route to its RIPng neighbors.

When configuring the Foundry device to redistribute routes, you can optionally specify a metric for the redistributed routes. If you do not explicitly configure a metric, the default metric value of one is used.

For example, to redistribute OSPF V3 routes into RIPng, enter the following command:

```
FastIron(config)# ipv6 router rip
FastIron(config-ripng-router)# redistribute ospf
```

Syntax: redistribute bgp | connected | isis | ospf | static [metric <number>]

For the metric, specify a numerical value that is consistent with RIPng.

Controlling Distribution of Routes Via RIPng

You can create a prefix list and then apply it to RIPng routing updates that are received or sent on a router interface. Performing this task allows you to control the distribution of routes via RIPng.

For example, to permit the inclusion of routes with the prefix 2001::/16 in RIPng routing updates sent from Ethernet interface 3/1, enter the following commands:

```
FastIron(config)# ipv6 prefix-list routesfor2001 permit 2001::/16
FastIron(config)# ipv6 router rip
FastIron(config-ripng-router)# distribute-list prefix-list routesfor2001 out
ethernet 3/1
```

To deny prefix lengths greater than 64 bits in routes that have the prefix 3EE0:A99::/64 and allow all other routes received on tunnel interface 3/1, enter the following commands:

```
FastIron(config)# ipv6 prefix-list 3ee0routes deny 3ee0:a99::/64 le 128
FastIron(config)# ipv6 prefix-list 3ee0routes permit ::/0 ge 0 le 128
FastIron(config)# ipv6 router rip
FastIron(config-ripng-router)# distribute-list prefix-list 3ee0routes in
tunnel 1
```

Syntax: [no] distribute-list prefix-list <name> in | out <interface> <port>

The <name> parameter indicates the name of the prefix list generated using the **ipv6 prefix-list** command.

The **in** keyword indicates that the prefix list is applied to incoming routing updates on the specified interface.

The **out** keyword indicates that the prefix list is applied to outgoing routing updates on the specified interface.

For the <interface> parameter, you can specify the **ethernet**, **loopback**, **ve**, or **tunnel** keywords. If you specify an Ethernet interface, also specify the port number associated with the interface. If you specify a VE or tunnel interface, also specify the VE or tunnel number.

To remove the distribution list, use the **no** form of this command.

Configuring Poison Reverse Parameters

By default, poison reverse is disabled on a RIPng router. If poison reverse is enabled, RIPng advertises routes it learns from a particular interface over that same interface with a metric of 16, which means that the route is unreachable.

If poison reverse is enabled on the RIPng router, it takes precedence over split horizon (if it is also enabled).

To enable poison reverse on the RIPng router, enter the following commands:

```
FastIron(config)# ipv6 router rip
FastIron(config-ripng-router)# poison-reverse
```

Syntax: [no] poison-reverse

To disable poison-reverse, use the **no** version of this command.

By default, if a RIPng interface goes down, the Foundry device does not send a triggered update for the interface's IPv6 networks.

To better handle this situation, you can configure a RIPng router to send a triggered update containing the local routes of the disabled interface with an unreachable metric of 16 to the other RIPng routers in the routing domain. You can enable the sending of a triggered update by entering the following commands:

```
FastIron(config)# ipv6 router rip
FastIron(config-ripng-router)# poison-local-routes
```

Syntax: [no] poison-local-routes

To disable the sending of a triggered update, use the **no** version of this command.

Clearing RIPng Routes from IPv6 Route Table

To clear all RIPng routes from the RIPng route table and the IPv6 main route table and reset the routes, enter the following command at the Privileged EXEC level or any of the Config levels of the CLI:

```
FastIron# clear ipv6 rip routes
```

Syntax: clear ipv6 rip routes

Displaying RIPng Information

You can display the following RIPng information:

- RIPng configuration
- RIPng routing table

Displaying RIPng Configuration

To display RIPng configuration information, enter the following command at any CLI level:

```
FastIron# show ipv6 rip
 IPv6 rip enabled, port 521
     Administrative distance is 120
     Updates every 30 seconds, expire after 180
     Holddown lasts 180 seconds, garbage collect after 120
     Split horizon is on; poison reverse is off
     Default routes are not generated
     Periodic updates 0, trigger updates 0
     Distribute List, Inbound : Not set
     Distribute List, Outbound : Not set
     Redistribute: CONNECTED
```

Syntax: show ipv6 rip

This display shows the following information:

Table 34.2:RIPng configuration fields

This Field...	Displays...
IPv6 RIP status/port	The status of RIPng on the Foundry device. Possible status is “enabled” or “disabled.” The UDP port number over which RIPng is enabled.
Administrative distance	The setting of the administrative distance for RIPng.
Updates/expiration	The settings of the RIPng update and timeout timers.
Holddown/garbage collection	The settings of the RIPng hold-down and garbage-collection timers.

Table 34.2:RIPng configuration fields

This Field...	Displays...
Split horizon/poison reverse	The status of the RIPng split horizon and poison reverse features. Possible status is “on” or “off.”
Default routes	The status of RIPng default routes.
Periodic updates/trigger updates	The number of periodic updates and triggered updates sent by the RIPng router.
Distribution lists	The inbound and outbound distribution lists applied to RIPng.
Redistribution	The types of IPv6 routes redistributed into RIPng. The types can include the following: • STATIC – IPv6 static routes are redistributed into RIPng. • CONNECTED – Directly connected IPv6 networks are redistributed into RIPng. • OSPF – OSPF V3 routes are redistributed into RIPng.

Displaying RIPng Routing Table

To display the RIPng routing table, enter the following command at any CLI level:

```
FastIron# show ipv6 rip route
IPv6 RIP Routing Table - 4 entries:
 2000:4::/64, from ::, null   (0)
          CONNECTED, metric 1, tag 0, timers: none
 2002:c0a8:46a::/64, from ::, null   (1)
          CONNECTED, metric 1, tag 0, timers: none
 2999::1/128, from ::, null   (2)
          CONNECTED, metric 1, tag 0, timers: none
 5000:2::/64, from ::, null   (3)
          CONNECTED, metric 1, tag 0, timers: none
```

Syntax: show ipv6 rip route [<ipv6-prefix>/<prefix-length> | <ipv6-address>]

The <ipv6-prefix>/<prefix-length> parameters restrict the display to the entries for the specified IPv6 prefix. You must specify the <ipv6-prefix> parameter in hexadecimal using 16-bit values between colons as documented in RFC 2373. You must specify the <prefix-length> parameter as a decimal value. A slash mark (/) must follow the <ipv6-prefix> parameter and precede the <prefix-length> parameter.

The <ipv6-address> parameter restricts the display to the entries for the specified IPv6 address. You must specify this parameter in hexadecimal using 16-bit values between colons as documented in RFC 2373.

This display shows the following information:

Table 34.3:RIPng routing table fields

This Field...	Displays...
RIPng Routing Table entries	The total number of entries in the RIPng routing table.
<ipv6-prefix>/<prefix-length>	The IPv6 prefix and prefix length.
<ipv6-address>	The IPv6 address.

Table 34.3:RIPng routing table fields

This Field...	Displays...
Next-hop router	The next-hop router for this Foundry device. If:: appears, the route is originated locally.
Interface	The interface name. If “null” appears, the interface is originated locally.
Source of route	The source of the route information. The source can be one of the following: • RIP – routes learned by RIPng. • CONNECTED – IPv6 routes redistributed from directly connected networks. • STATIC – IPv6 static routes are redistributed into RIPng. • OSPF – OSPF V3 routes are redistributed into RIPng.
Metric <number>	The cost of the route. The <number> parameter indicates the number of hops to the destination.
Tag <number>	The tag value of the route.
Timers:	Indicates if the hold-down timer or the garbage-collection timer is set.

Chapter 35

Configuring OSPF Version 2 (IPv4)

This chapter describes how to configure OSPF Version 2 on Foundry Layer 3 Switches using the CLI. OSPF Version 2 is supported on devices running IPv4.

NOTE: The terms *Layer 3 Switch* and *router* are used interchangeably in this chapter and mean the same thing.

Overview of OSPF

OSPF is a link-state routing protocol. The protocol uses link-state advertisements (LSA) to update neighboring routers regarding its interfaces and information on those interfaces. The router floods these LSAs to all neighboring routers to update them regarding the interfaces. Each router maintains an identical database that describes its area topology to help a router determine the shortest path between it and any neighboring router.

Foundry Layer 3 Switches support the following types of LSAs, which are described in RFC 1583:

- Router link
- Network link
- Summary link
- Autonomous system (AS) summary link
- AS external link
- Not-So-Stubby Area (NSSA) external link

OSPF is built upon a hierarchy of network components. The highest level of the hierarchy is the ***Autonomous System (AS)***. An autonomous system is defined as a number of networks, all of which share the same routing and administration characteristics.

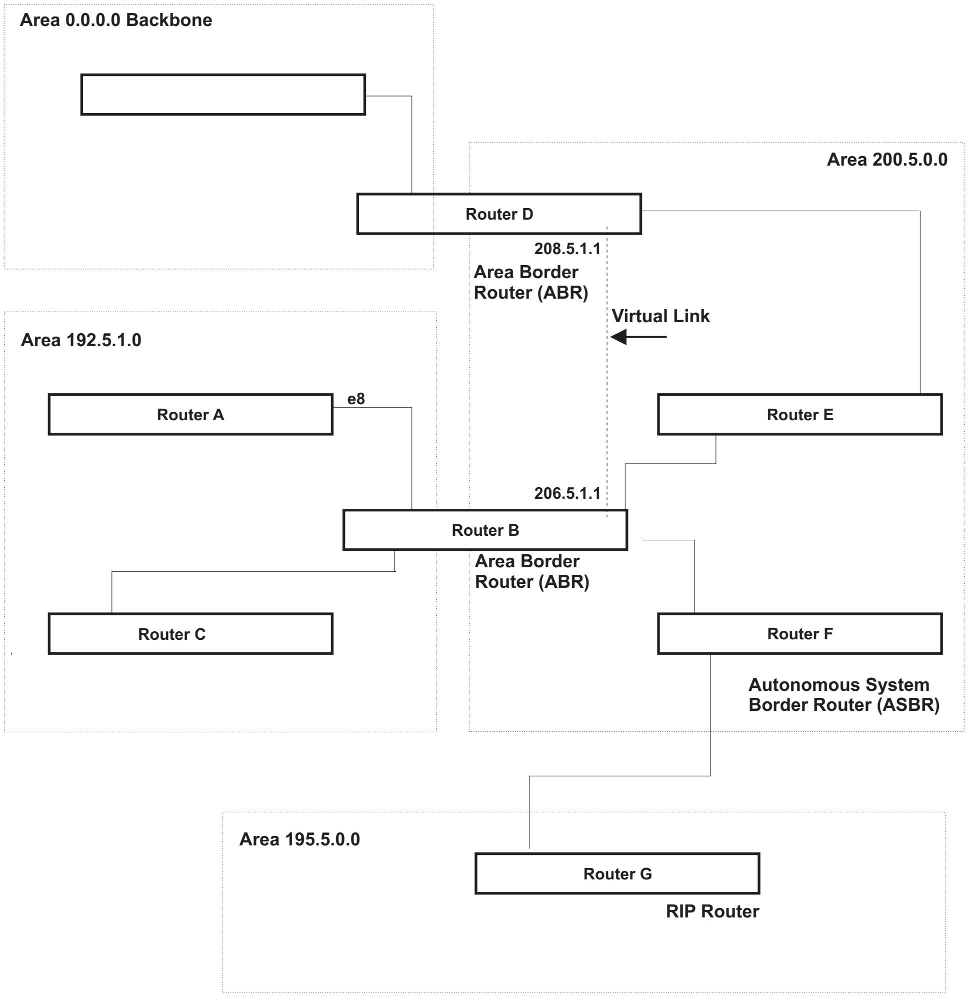
An AS can be divided into multiple ***areas*** as shown in Figure 35.1 on page 35-2. Each area represents a collection of contiguous networks and hosts. Areas limit the area to which link-state advertisements are broadcast, thereby limiting the amount of flooding that occurs within the network. An area is represented in OSPF by either an IP address or a number.

You can further limit the broadcast area of flooding by defining an area range. The area range allows you to assign an aggregate value to a range of IP addresses. This aggregate value becomes the address that is advertised instead all of the individual addresses it represents being advertised. You can assign up to 32 ranges in an OSPF area.

An OSPF router can be a member of multiple areas. Routers with membership in multiple areas are known as ***Area Border Routers (ABRs)***. Each ABR maintains a separate topological database for each area the router is in. Each topological database contains all of the LSA databases for each router within a given area. The routers within the same area have identical topological databases. The ABR is responsible for forwarding routing information or changes between its border areas.

An ***Autonomous System Boundary Router (ASBR)*** is a router that is running multiple protocols and serves as a gateway to routers outside an area and those operating with different protocols. The ASBR is able to import and translate different protocol routes into OSPF through a process known as ***redistribution***. For more details on redistribution and configuration examples, see “Enable Route Redistribution” on page 35-27.

Figure 35.1 OSPF Operating in a Network



OSPF Point-to-Point Links

Platform Support:

- FESX devices running software release 02.2.00 and later
- FSX devices running software release 02.3.01 and later

One important OSPF process is ***Adjacency***. Adjacency occurs when a relationship is formed between neighboring routers for the purpose of exchanging routing information. Adjacent OSPF neighbor routers go beyond the simple Hello packet exchange; they exchange database information. In order to minimize the amount of information exchanged on a particular segment, one of the first steps in creating adjacency is to assign a Designated Router (DR) and a Backup Designated Router (BDR). The Designated Router ensures that there is a central point of contact, thereby improving convergence time within a multi-access segment.

In an OSPF point-to-point network, where a direct Layer 3 connection exists between a single pair of OSPF routers, there is no need for Designated and Backup Designated Routers, as is the case in OSPF multi-access networks. Without the need for Designated and Backup Designated routers, a point-to-point network establishes adjacency and converges faster. The neighboring routers become adjacent whenever they can communicate directly. In contrast, in broadcast and non-broadcast multi-access (NBMA) networks, the Designated Router and Backup Designated Router become adjacent to all other routers attached to the network.

To configure an OSPF point-to-point link, see “Configuring an OSPF Point-to-Point Link” on page 35-35.

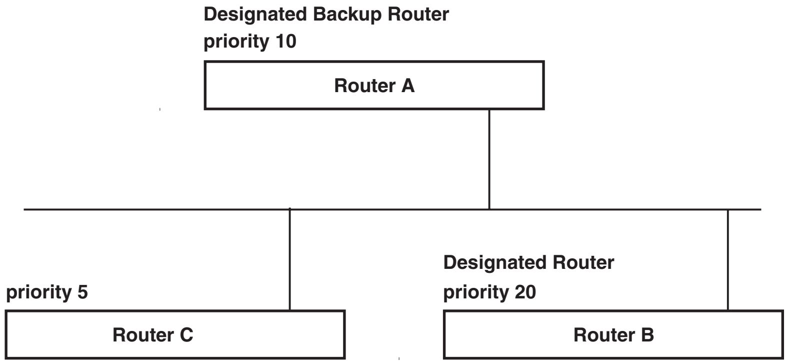
Designated Routers in Multi-Access Networks

In a network that has multiple routers attached, OSPF elects one router to serve as the designated router (DR) and another router on the segment to act as the backup designated router (BDR). This arrangement minimizes the amount of repetitive information that is forwarded on the network by forwarding all messages to the designated router and backup designated routers responsible for forwarding the updates throughout the network.

Designated Router Election in Multi-Access Networks

In a network with no designated router and no backup designated router, the neighboring router with the highest priority is elected as the DR, and the router with the next largest priority is elected as the BDR, as shown in Figure 35.2

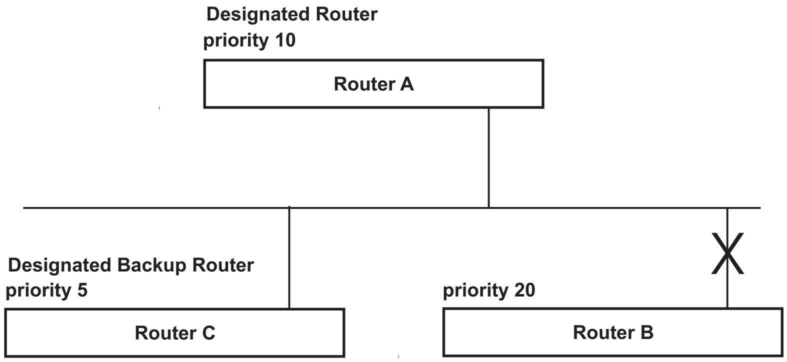
Figure 35.2 Designated and Backup Router Election



If the DR goes off-line, the BDR automatically becomes the DR. The router with the next highest priority becomes the new BDR. This process is shown in Figure 35.3.

NOTE: Priority is a configurable option at the interface level. You can use this parameter to help bias one router as the DR.

Figure 35.3 Backup Designated Router Becomes Designated Router



If two neighbors share the same priority, the router with the highest router ID is designated as the DR. The router with the next highest router ID is designated as the BDR.

NOTE: By default, the Foundry router ID is the IP address configured on the lowest numbered loopback interface. If the Layer 3 Switch does not have a loopback interface, the default router ID is the lowest numbered IP address configured on the device. For more information or to change the router ID, see "Changing the Router ID" on page 29-24.

When multiple routers on the same network are declaring themselves as DRs, then both priority and router ID are used to select the designated router and backup designated routers.

When only one router on the network claims the DR role despite neighboring routers with higher priorities or router IDs, this router remains the DR. This is also true for BDRs.

The DR and BDR election process is performed when one of the following events occurs:

- An interface is in a waiting state and the wait time expires
- An interface is in a waiting state and a hello packet is received that addresses the BDR
- A change in the neighbor state occurs, such as:
  - A neighbor state transitions from 2 or higher
  - Communication to a neighbor is lost
  - A neighbor declares itself to be the DR or BDR for the first time

OSPF RFC 1583 and 2178 Compliance

Foundry routers are configured, by default, to be compliant with the RFC 1583 OSPF V2 specification. Foundry routers can also be configured to operate with the latest OSPF standard, RFC 2178.

NOTE: For details on how to configure the system to operate with the RFC 2178, see "Modify OSPF Standard Compliance Setting" on page 35-35.

Reduction of Equivalent AS External LSAs

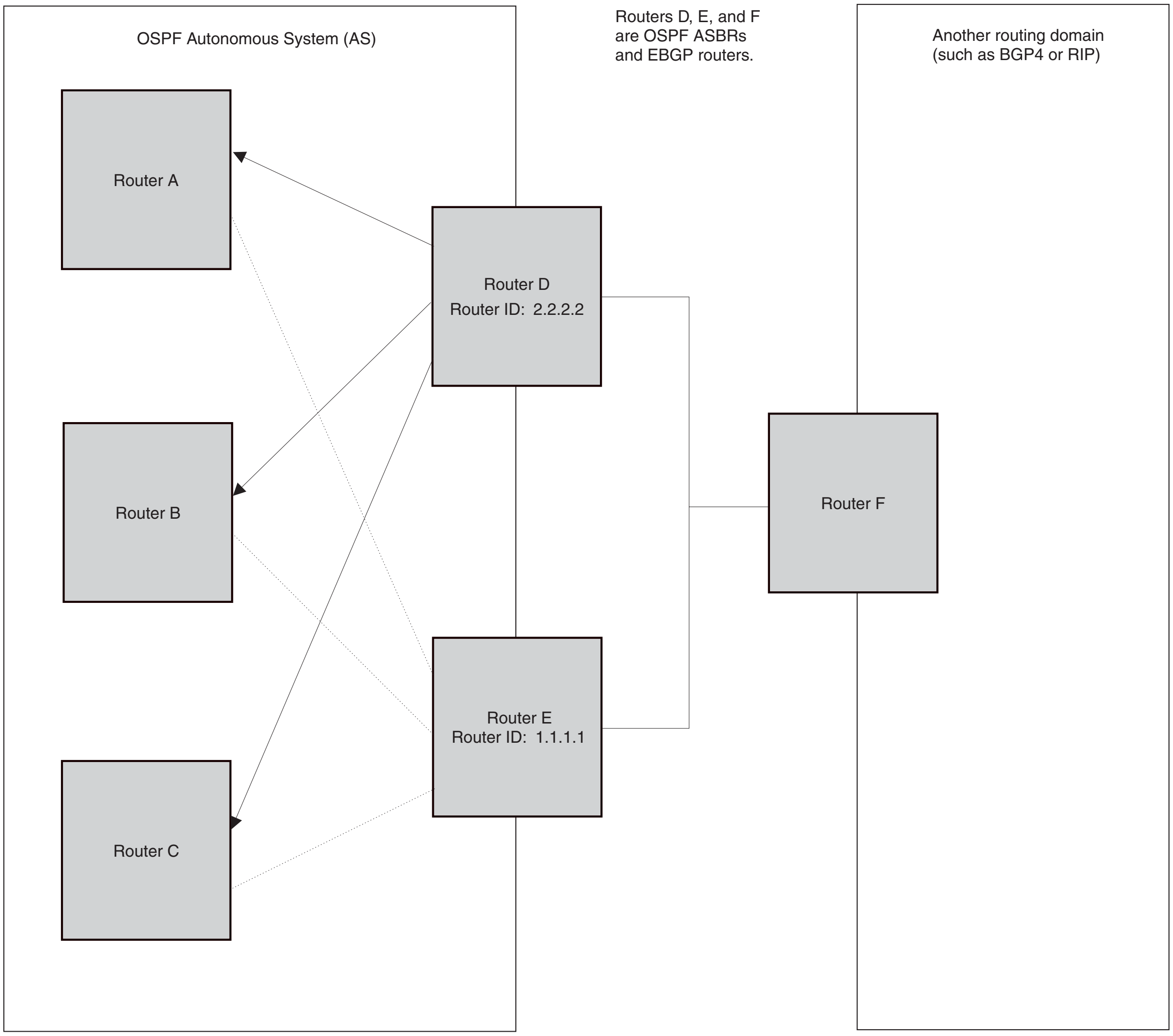
An OSPF ASBR uses AS External link advertisements (AS External LSAs) to originate advertisements of a route to another routing domain, such as a BGP4 or RIP domain. The ASBR advertises the route to the external domain by flooding AS External LSAs to all the other OSPF routers (except those inside stub networks) within the local OSPF Autonomous System (AS).

In some cases, multiple ASBRs in an AS can originate equivalent LSAs. The LSAs are equivalent when they have the same cost, the same next hop, and the same destination. Foundry devices optimize OSPF by eliminating duplicate AS External LSAs in this case. The Layer 3 Switch with the lower router ID flushes the duplicate External LSAs from its database and thus does not flood the duplicate External LSAs into the OSPF AS. AS External LSA reduction therefore reduces the size of the Layer 3 Switch's link state database.

This enhancement implements the portion of RFC 2328 that describes AS External LSA reduction. This enhancement is enabled by default, requires no configuration, and cannot be disabled.

Figure 35.4 shows an example of the AS External LSA reduction feature. In this example, Foundry Layer 3 Switches D and E are OSPF ASBRs, and thus communicate route information between the OSPF AS, which contains Routers A, B, and C, and another routing domain, which contains Router F. The other routing domain is running another routing protocol, such as BGP4 or RIP. Routers D, E, and F, therefore, are each running both OSPF and either BGP4 or RIP.

Figure 35.4 AS External LSA Reduction


OSPF Autonomous System (AS)
Routers D, E, and F are OSPF ASBRs and EBGP routers.
Another routing domain (such as BGP4 or RIP)
Router A
Router D
Router ID: 2.2.2.2
Router B
Router F
Router E
Router ID: 1.1.1.1
Router C

Notice that both Router D and Router E have a route to the other routing domain through Router F. In earlier software releases, if Routers D and E have equal-cost routes to Router F, then both Router D and Router E flood AS External LSAs to Routers A, B, and C advertising the route to Router F. Since both routers are flooding equivalent routes, Routers A, B, and C receive multiple routes with the same cost to the same destination (Router F). For Routers A, B, and C, either route to Router F (through Router D or through Router E) is equally good.

OSPF eliminates the duplicate AS External LSAs. When two or more Foundry Layer 3 Switches configured as ASBRs have equal-cost routes to the same next-hop router in an external routing domain, the ASBR with the highest router ID floods the AS External LSAs for the external domain into the OSPF AS, while the other ASBRs flush the equivalent AS External LSAs from their databases. As a result, the overall volume of route advertisement traffic within the AS is reduced and the Layer 3 Switches that flush the duplicate AS External LSAs have more memory for other OSPF data. In Figure 35.4, since Router D has a higher router ID than Router E, Router D floods the AS External LSAs for Router F to Routers A, B, and C. Router E flushes the equivalent AS External LSAs from its database.

Algorithm for AS External LSA Reduction

Figure 35.4 shows an example in which the normal AS External LSA reduction feature is in effect. The behavior changes under the following conditions:

- There is one ASBR advertising (originating) a route to the external destination, but one of the following happens:
  - A second ASBR comes on-line
  - A second ASBR that is already on-line begins advertising an equivalent route to the same destination.

  In either case above, the router with the higher router ID floods the AS External LSAs and the other router flushes its equivalent AS External LSAs. For example, if Router D is offline, Router E is the only source for a route to the external routing domain. When Router D comes on-line, it takes over flooding of the AS External LSAs to Router F, while Router E flushes its equivalent AS External LSAs to Router F.
- One of the ASBRs starts advertising a route that is no longer equivalent to the route the other ASBR is advertising. In this case, the ASBRs each flood AS External LSAs. Since the LSAs either no longer have the same cost or no longer have the same next-hop router, the LSAs are no longer equivalent, and the LSA reduction feature no longer applies.
- The ASBR with the higher router ID becomes unavailable or is reconfigured so that it is no longer an ASBR. In this case, the other ASBR floods the AS External LSAs. For example, if Router D goes off-line, then Router E starts flooding the AS with AS External LSAs for the route to Router F.

Support for OSPF RFC 2328 Appendix E

Foundry devices provide support for Appendix E in OSPF RFC 2328. Appendix E describes a method to ensure that an OSPF router (such as aFoundry Layer 3 Switch) generates unique link state IDs for type-5 (External) link state advertisements (LSAs) in cases where two networks have the same network address but different network masks.

NOTE: Support for Appendix E of RFC 2328 is enabled automatically and cannot be disabled. No user configuration is required.

Normally, an OSPF router uses the network address alone for the link state ID of the link state advertisement (LSA) for the network. For example, if the router needs to generate an LSA for network 10.1.2.3 255.0.0.0, the router generates ID 10.1.2.3 for the LSA.

However, suppose that an OSPF router needs to generate LSAs for all the following networks:

- 10.0.0.0 255.0.0.0
- 10.0.0.0 255.255.0.0
- 10.0.0.0 255.255.255.0

All three networks have the same network address, 10.0.0.0. Without support for RFC 2328 Appendix E, an OSPF router uses the same link state ID, 10.0.0.0, for the LSAs for all three networks. For example, if the router generates an LSA with ID 10.0.0.0 for network 10.0.0.0 255.0.0.0, this LSA conflicts with the LSA generated for network 10.0.0.0 255.255.0.0 or 10.0.0.0 255.255.255.0. The result is multiple LSAs that have the same ID but that contain different route information.

When appendix E is supported, the router generates the link state ID for a network as follows:

1. Does an LSA with the network address as its ID already exist?

  - No – Use the network address as the ID.
  - Yes – Go to Step 2.

2. Compare the networks that have the same network address, to determine which network is more specific. The more specific network is the one that has more contiguous one bits in its network mask. For example, network 10.0.0.0 255.255.0.0 is more specific than network 10.0.0.0 255.0.0.0, because the first network has 16 ones bits (255.255.0.0) whereas the second network has only 8 ones bits (255.0.0.0).
   - For the less specific network, use the networks address as the ID.
   - For the more specific network, use the network's broadcast address as the ID. The broadcast address is the network address, with all ones bits in the host portion of the address. For example, the broadcast address for network 10.0.0.0 255.255.0.0 is 10.0.0.255.

   If this comparison results in a change to the ID of an LSA that has already been generated, the router generates a new LSA to replace the previous one. For example, if the router has already generated an LSA for network with ID 10.0.0.0 for network 10.0.0.0 255.255.255.0, the router must generate a new LSA for the network, if the router needs to generate an LSA for network 10.0.0.0 255.255.0.0 or 10.0.0.0 255.0.0.0.

Dynamic OSPF Activation and Configuration

OSPF is automatically activated when you enable it. The protocol does not require a software reload.

You can configure and save the following OSPF changes without resetting the system:

- All OSPF interface-related parameters (for example: area, hello timer, router dead time cost, priority, re-transmission time, transit delay)
- All area parameters
- All area range parameters
- All virtual-link parameters
- All global parameters
- Creation and deletion of an area, interface or virtual link

In addition, you can make the following changes without a system reset by first disabling and then re-enabling OSPF operation:

- Changes to address ranges
- Changes to global values for redistribution
- Addition of new virtual links

You also can change the amount of memory allocated to various types of LSA entries. However, these changes require a system reset or reboot.

Dynamic OSPF Memory

FastIron devices dynamically allocate memory for Link State Advertisements (LSAs) and other OSPF data structures. This eliminates overflow conditions and does not require a reload to change OSPF memory allocation. So long as the Layer 3 Switch has free (unallocated) dynamic memory, OSPF can use the memory.

To display the current allocations of dynamic memory, enter the show memory command. See the *Foundry Switch and Router Command Line Interface Reference*.

Configuring OSPF

To begin using OSPF on the router, perform the steps outlined below:

1. Enable OSPF on the router.
2. Assign the areas to which the router will be attached.

3. Assign individual interfaces to the OSPF areas.
4. Define redistribution filters, if desired.
5. Enable redistribution, if you defined redistribution filters.
6. Modify default global and port parameters as required.
7. Modify OSPF standard compliance, if desired.

NOTE: OSPF is automatically enabled without a system reset.

Configuration Rules

- Foundry FastIron devices support a maximum of 676 OSPF interfaces.
- If a router is to operate as an ASBR, you must enable the ASBR capability at the system level.
- Redistribution must be enabled on routers configured to operate as ASBRs.
- All router ports must be assigned to one of the defined areas on an OSPF router. When a port is assigned to an area, all corresponding subnets on that port are automatically included in the assignment.

OSPF Parameters

You can modify or set the following global and interface OSPF parameters.

Global Parameters

- Modify OSPF standard compliance setting.
- Assign an area.
- Define an area range.
- Define the area virtual link.
- Set global default metric for OSPF.
- Change the reference bandwidth for the default cost of OSPF interfaces.
- Disable or re-enable load sharing.
- Enable or disable default-information-originate.
- Modify Shortest Path First (SPF) timers
- Define external route summarization
- Define redistribution metric type.
- Define deny redistribution.
- Define permit redistribution.
- Enable redistribution.
- Change the LSA pacing interval.
- Modify OSPF Traps generated.
- Modify database overflow interval.

Interface Parameters

- Assign interfaces to an area.
- Define the authentication key for the interface.
- Change the authentication-change interval
- Modify the cost for a link.

- Modify the dead interval.
- Modify MD5 authentication key parameters.
- Modify the priority of the interface.
- Modify the retransmit interval for the interface.
- Modify the transit delay of the interface.

NOTE: When using the CLI, you set global level parameters at the OSPF CONFIG Level of the CLI. To reach that level, enter **router ospf…** at the global CONFIG Level. Interface parameters for OSPF are set at the interface CONFIG Level using the CLI command, **ip ospf…**

When using the Web management interface, you set OSPF global parameters using the OSPF configuration panel. All other parameters are accessed through links accessed from the OSPF configuration sheet.

Enable OSPF on the Router

When you enable OSPF on the router, the protocol is automatically activated. To enable OSPF on the router, enter the following CLI command:

```
FastIron(config)#router ospf
```

This command launches you into the OSPF router level where you can assign areas and modify OSPF global parameters.

Note Regarding Disabling OSPF

If you disable OSPF, the Layer 3 Switch removes all the configuration information for the disabled protocol from the running-config. Moreover, when you save the configuration to the startup-config file after disabling one of these protocols, all the configuration information for the disabled protocol is removed from the startup-config file.

NOTE: If you are running software release 02.4.00 or later and do not want to delete the OSPF configuration information, use the CLI command **clear ip ospf process** instead of **no router ospf**. See “Resetting OSPF” on page 35-9.

When you enter the **no router ospf** command, the CLI displays a warning message such as the following:

```
FastIron(config-ospf-router)#no router ospf
router ospf mode now disabled. All ospf config data will be lost when writing to
flash!
```

The Web management interface does not display a warning message.

If you have disabled the protocol but have not yet saved the configuration to the startup-config file and reloaded the software, you can restore the configuration information by re-entering the command to enable the protocol (ex: **router ospf**), or by selecting the Web management option to enable the protocol. If you have already saved the configuration to the startup-config file and reloaded the software, the information is gone.

If you are testing an OSPF configuration and are likely to disable and re-enable the protocol, you might want to make a backup copy of the startup-config file containing the protocol’s configuration information. This way, if you remove the configuration information by saving the configuration after disabling the protocol, you can restore the configuration by copying the backup copy of the startup-config file onto the flash memory.

Resetting OSPF

Platform Support:

- FESX/FSX/FWSX devices running software release 02.4.00 and later

The **clear ip ospf process** command globally resets (disables then re-enables) OSPF without deleting the OSPF configuration information. This command is equivalent to entering the commands **no router ospf** followed by **router ospf**. Whereas the **no router ospf** command disables OSPF and removes all the configuration information for the disabled protocol from the running-config, the **router ospf** command re-enables OSPF and restores the OSPF configuration information.

The **clear ip ospf process** command is useful If you are testing an OSPF configuration and are likely to disable and re-enable the protocol. This way, you do not have to save the configuration after disabling the protocol, and you do not have to restore the configuration by copying the backup copy of the startup-config file onto the flash memory.

To reset OSPF without deleting the OSPF configuration, enter the following command at the Global CONFIG level or at the Router OSPF level of the CLI:

```
FastIron#clear ip ospf process
```

Syntax: clear ip ospf process

Assign OSPF Areas

Once OSPF is enabled on the system, you can assign areas. Assign an IP address or number as the ***area ID*** for each area. The area ID is representative of all IP addresses (subnets) on a router port. Each port on a router can support one area.

An area can be ***normal***, a ***stub***, or a ***Not-So-Stubby Area (NSSA)***.

- Normal – OSPF routers within a normal area can send and receive External Link State Advertisements (LSAs).
- Stub – OSPF routers within a stub area cannot send or receive External LSAs. In addition, OSPF routers in a stub area must use a default route to the area’s Area Border Router (ABR) or Autonomous System Boundary Router (ASBR) to send traffic out of the area.
- NSSA – The ASBR of an NSSA can import external route information into the area.
  - ASBRs redistribute (import) external routes into the NSSA as type 7 LSAs. Type-7 External LSAs are a special type of LSA generated only by ASBRs within an NSSA, and are flooded to all the routers within only that NSSA.
  - ABRs translate type 7 LSAs into type 5 External LSAs, which can then be flooded throughout the AS. You can configure address ranges on the ABR of an NSSA so that the ABR converts multiple type-7 External LSAs received from the NSSA into a single type-5 External LSA.

    When an NSSA contains more than one ABR, OSPF elects one of the ABRs to perform the LSA translation for NSSA. OSPF elects the ABR with the highest router ID. If the elected ABR becomes unavailable, OSPF automatically elects the ABR with the next highest router ID to take over translation of LSAs for the NSSA. The election process for NSSA ABRs is automatic.

EXAMPLES:

To set up the OSPF areas shown in Figure 35.1 on page 35-2, enter the following commands.

```
FastIron(config-ospf-router)#area 192.5.1.0
FastIron(config-ospf-router)#area 200.5.0.0
FastIron(config-ospf-router)#area 195.5.0.0
FastIron(config-ospf-router)#area 0.0.0.0
FastIron(config-ospf-router) write memory
```

Syntax: area <num> | <ip-addr>

The <num> | <ip-addr> parameter specifies the area number, which can be a number or in IP address format. If you specify an number, the number can be from 0 – 2,147,483,647.

NOTE: You can assign one area on a router interface. For example, if the system or chassis module has 16 ports, 16 areas are supported on the chassis or module.

Assign a Totally Stubby Area

By default, the Layer 3 Switch sends summary LSAs (LSA type 3) into stub areas. You can further reduce the number of link state advertisements (LSAs) sent into a stub area by configuring the Layer 3 Switch to stop sending summary LSAs (type 3 LSAs) into the area. You can disable the summary LSAs when you are configuring the stub area or later after you have configured the area.

This feature disables origination of summary LSAs, but the Layer 3 Switch still accepts summary LSAs from OSPF neighbors and floods them to other neighbors. The Layer 3 Switch can form adjacencies with other routers regardless of whether summarization is enabled or disabled for areas on each router.

When you enter a command or apply a Web management option to disable the summary LSAs, the change takes effect immediately. If you apply the option to a previously configured area, the Layer 3 Switch flushes all of the summary LSAs it has generated (as an ABR) from the area.

NOTE: This feature applies only when the Layer 3 Switch is configured as an Area Border Router (ABR) for the area. To completely prevent summary LSAs from being sent to the area, disable the summary LSAs on each OSPF router that is an ABR for the area.

This feature does not apply to Not So Stubby Areas (NSSAs).

To disable summary LSAs for a stub area, enter commands such as the following:

```
FastIron(config-ospf-router)#area 40 stub 99 no-summary
```

Syntax: area <num> | <ip-addr> stub <cost> [no-summary]

The <num> | <ip-addr> parameter specifies the area number, which can be a number or in IP address format. If you specify a number, the number can be from 0 – 2,147,483,647.

The **stub** <cost> parameter specifies an additional cost for using a route to or from this area and can be from 1 – 16777215. There is no default. Normal areas do not use the cost parameter.

The **no-summary** parameter applies only to stub areas and disables summary LSAs from being sent into the area.

NOTE: You can assign one area on a router interface. For example, if the system or chassis module has 16 ports, 16 areas are supported on the chassis or module.

Assign a Not-So-Stubby Area (NSSA)

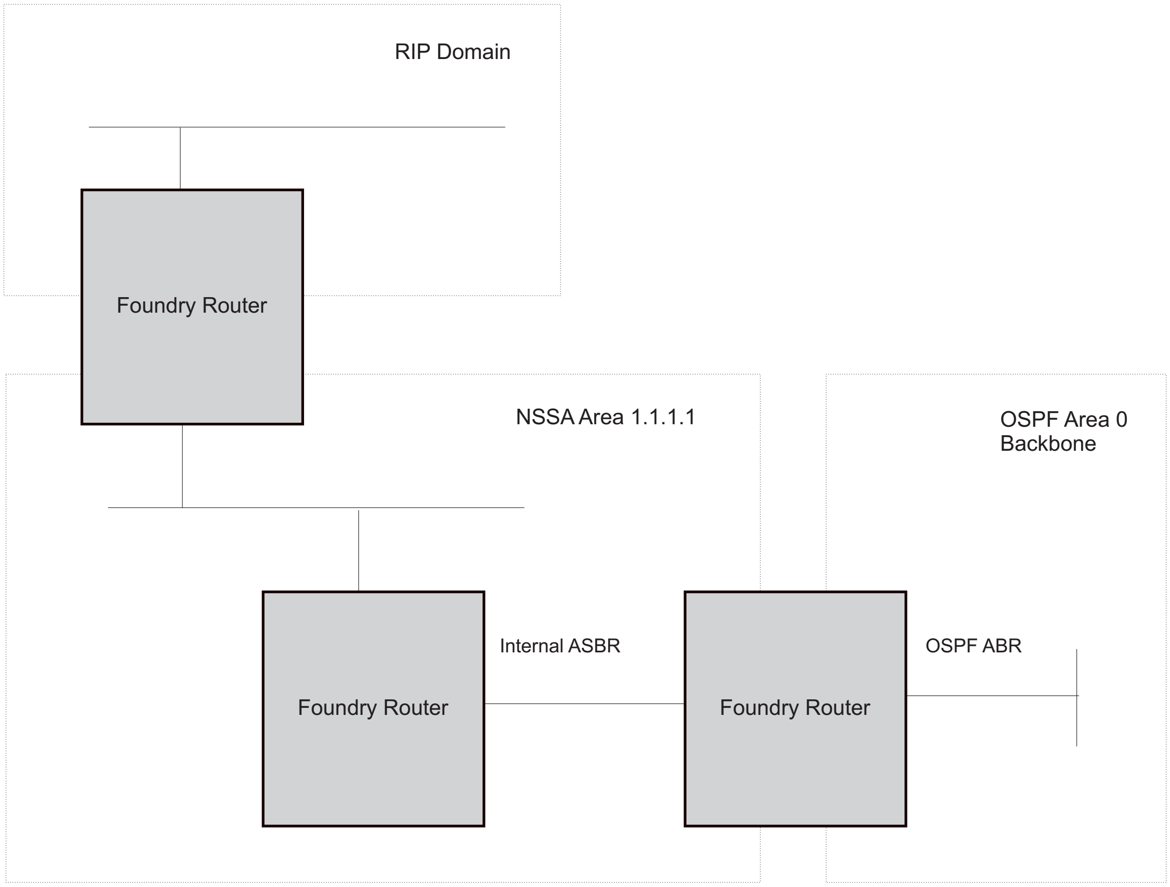
The OSPF Not So Stubby Area (NSSA) feature enables you to configure OSPF areas that provide the benefits of stub areas, but that also are capable of importing external route information. OSPF does not flood external routes from other areas into an NSSA, but does translate and flood route information from the NSSA into other areas such as the backbone.

NSSAs are especially useful when you want to summarize Type-5 External LSAs (external routes) before forwarding them into an OSPF area. The OSPF specification (RFC 2328) prohibits summarization of Type-5 LSAs and requires OSPF to flood Type-5 LSAs throughout a routing domain. When you configure an NSSA, you can specify an address range for aggregating the external routes that the NSSA's ABR exports into other areas.

The Foundry implementation of NSSA is based on RFC 1587.

Figure 35.5 shows an example of an OSPF network containing an NSSA.

Figure 35.5 OSPF Network Containing an NSSA



This example shows two routing domains, a RIP domain and an OSPF domain. The ASBR inside the NSSA imports external routes from RIP into the NSSA as Type-7 LSAs, which the ASBR floods throughout the NSSA.

The ABR translates the Type-7 LSAs into Type-5 LSAs. If an area range is configured for the NSSA, the ABR also summarizes the LSAs into an aggregate LSA before flooding the Type-5 LSA(s) into the backbone.

Since the NSSA is partially “stubby” the ABR does not flood external LSAs from the backbone into the NSSA. To provide access to the rest of the Autonomous System (AS), the ABR generates a default Type-7 LSA into the NSSA.

Configuring an NSSA

To configure OSPF area 1.1.1.1 as an NSSA, enter the following commands.

```
FastIron(config)#router ospf
FastIron(config-ospf-router)#area 1.1.1.1 nssa 1
FastIron(config-ospf-router)#write memory
```

Syntax: area <num> | <ip-addr> nssa <cost> | default-information-originate

The <num> | <ip-addr> parameter specifies the area number, which can be a number or in IP address format. If you specify an number, the number can be from 0 – 2,147,483,647.

The **nssa** <cost> | **default-information-originate** parameter specifies that this is a Not-So-Stubby-Area (NSSA). The <cost> specifies an additional cost for using a route to or from this NSSA and can be from 1 – 16777215. There is no default. Normal areas do not use the cost parameter. Alternatively, you can use the **default-information-originate** parameter causes the Layer 3 Switch to inject the default route into the NSSA.

NOTE: The Layer 3 Switch does not inject the default route into an NSSA by default.

NOTE: You can assign one area on a router interface. For example, if the system or chassis module has 16 ports, 16 areas are supported on the chassis or module.

To configure additional parameters for OSPF interfaces in the NSSA, use the **ip ospf area…** command at the interface level of the CLI.

Configuring a Summary Address for the NSSA

If you want the ABR that connects the NSSA to other areas to summarize the routes in the NSSA before translating them into Type-5 LSAs and flooding them into the other areas, configure a summary address. The ABR creates an aggregate value based on the summary address. The aggregate value becomes the address that the ABR advertises instead of advertising the individual addresses represented by the aggregate.

To configure a summary address in NSSA 1.1.1.1, enter the following commands. This example assumes that you have already configured NSSA 1.1.1.1.

```
FastIron(config)#router ospf
FastIron(config-ospf-router)#summary-address 209.157.22.1 255.255.0.0
FastIron(config-ospf-router)#write memory
```

Syntax: [no] summary address <ip-addr> <ip-mask>

The <ip-addr> parameter specifies the IP address portion of the range. The software compares the address with the significant bits in the mask. All network addresses that match this comparison are summarized in a single route advertised by the router.

The <ip-mask> parameter specifies the portions of the IP address that a route must contain to be summarized in the summary route. In the example above, all networks that begin with 209.157 are summarized into a single route.

Assigning an Area Range (optional)

You can assign a ***range*** for an area, but it is not required. Ranges allow a specific IP address and mask to represent a range of IP addresses within an area, so that only that reference range address is advertised to the network, instead of all the addresses within that range. Each area can have up to 32 range addresses.

EXAMPLES:

To define an area range for subnets on 193.45.5.1 and 193.45.6.2, enter the following command:

```
FastIron(config)#router ospf
FastIron(config-ospf-router)#area 192.45.5.1 range 193.45.0.0 255.255.0.0
FastIron(config-ospf-router)#area 193.45.6.2 range 193.45.0.0 255.255.0.0
```

Syntax: area <num> | <ip-addr> range <ip-addr> <ip-mask>

The <num> | <ip-addr> parameter specifies the area number, which can be in IP address format.

The **range** <ip-addr> parameter specifies the IP address portion of the range. The software compares the address with the significant bits in the mask. All network addresses that match this comparison are summarized in a single route advertised by the router.

The <ip-mask> parameter specifies the portions of the IP address that a route must contain to be summarized in the summary route. In the example above, all networks that begin with 193.45 are summarized into a single route.

Assigning Interfaces to an Area

Once you define OSPF areas, you can assign interfaces to the areas. All router ports must be assigned to one of the defined areas on an OSPF router. When a port is assigned to an area, all corresponding subnets on that port are automatically included in the assignment.

To assign interface 1/8 to area 192.5.0.0 and then save the changes, enter the following commands:

```
FastIron(config-ospf-router)#interface e 1/8
FastIron(config-if-1/8)#ip ospf area 192.5.0.0
FastIron(config-if-1/8)#write memory
```

Modify Interface Defaults

OSPF has interface parameters that you can configure. For simplicity, each of these parameters has a default value. No change to these default values is required except as needed for specific network configurations.

Port default values can be modified using the following commands at the interface configuration level of the CLI:

- ip ospf area <ip-addr>
- ip ospf auth-change-wait-time <secs>
- ip ospf authentication-key [0 | 1] <string>
- ip ospf cost <num>
- ip ospf dead-interval <value>
- ip ospf hello-interval <value>
- ip ospf md5-authentication key-activation-wait-time <num> | key-id <num> [0 | 1] key <string>
- ip ospf passive
- ip ospf priority <value>
- ip ospf retransmit-interval <value>
- ip ospf transmit-delay <value>

For a complete description of these parameters, see the summary of OSPF port parameters in the next section.

OSPF Interface Parameters

The following parameters apply to OSPF interfaces.

Area: Assigns an interface to a specific area. You can assign either an IP address or number to represent an OSPF Area ID. If you assign a number, it can be any value from 0 – 2,147,483,647.

Auth-change-wait-time: OSPF gracefully implements authentication changes to allow all routers to implement the change and thus prevent disruption to neighbor adjacencies. During the authentication-change interval, both the old and new authentication information is supported. The default authentication-change interval is 300 seconds (5 minutes). You change the interval to a value from 0 – 14400 seconds.

Authentication-key: OSPF supports three methods of authentication for each interface—none, simple password, and MD5. Only one method of authentication can be active on an interface at a time. The default authentication value is none, meaning no authentication is performed.

- The simple password method of authentication requires you to configure an alphanumeric password on an interface. The simple password setting takes effect immediately. All OSPF packets transmitted on the interface contain this password. Any OSPF packet received on the interface is checked for this password. If the password is not present, then the packet is dropped. The password can be up to eight characters long.
- The MD5 method of authentication requires you to configure a key ID and an MD5 Key. The key ID is a number from 1 – 255 and identifies the MD5 key that is being used. The MD5 key can be up to sixteen alphanumeric characters long.

Cost: Indicates the overhead required to send a packet across an interface. You can modify the cost to differentiate between 100 Mbps and 1000 Mbps (1 Gbps) links. The default cost is calculated by dividing 100 million by the bandwidth. For 10 Mbps links, the cost is 10. The cost for both 100 Mbps and 1000 Mbps links is 1, because the speed of 1000 Mbps was not in use at the time the OSPF cost formula was devised.

Dead-interval: Indicates the number of seconds that a neighbor router waits for a hello packet from the current router before declaring the router down. The value can be from 1 – 65535 seconds. The default is 40 seconds.

Hello-interval: Represents the length of time between the transmission of hello packets. The value can be from 1 – 65535 seconds. The default is 10 seconds.

MD5-authentication activation wait time: The number of seconds the Layer 3 Switch waits until placing a new MD5 key into effect. The wait time provides a way to gracefully transition from one MD5 key to another without disturbing the network. The wait time can be from 0 – 14400 seconds. The default is 300 seconds (5 minutes).

MD5-authentication key ID and key: A method of authentication that requires you to configure a key ID and an MD5 key. The key ID is a number from 1 – 255 and identifies the MD5 key that is being used. The MD5 key consists of up to 16 alphanumeric characters. The MD5 is encrypted and included in each OSPF packet transmitted.

Passive: When you configure an OSPF interface to be passive, that interface does not send or receive OSPF route updates. By default, all OSPF interfaces are active and thus can send and receive OSPF route information. Since a passive interface does not send or receive route information, the interface is in effect a stub network. OSPF interfaces are active by default.

NOTE: This option affects all IP subnets configured on the interface. If you want to disable OSPF updates only on some of the IP subnets on the interface, use the **ospf-ignore** or **ospf-passive** parameter with the **ip address** command. See “Assigning an IP Address to an Ethernet Port” on page 29-17.

Priority: Allows you to modify the priority of an OSPF router. The priority is used when selecting the designated router (DR) and backup designated routers (BDRs). The value can be from 0 – 255. The default is 1. If you set the priority to 0, the Layer 3 Switch does not participate in DR and BDR election.

Retransmit-interval: The time between retransmissions of link-state advertisements (LSAs) to adjacent routers for this interface. The value can be from 0 – 3600 seconds. The default is 5 seconds.

Transit-delay: The time it takes to transmit Link State Update packets on this interface. The value can be from 0 – 3600 seconds. The default is 1 second.

Encrypted Display of the Authentication String or MD5 Authentication Key

The optional **0 | 1** parameter with the **authentication-key** and **md5-authentication key-id** parameters affects encryption.

For added security, FastIron devices encrypt display of the password or authentication string. Encryption is enabled by default. The software also provides an optional parameter to disable encryption of a password or authentication string, on an individual OSPF area or OSPF interface basis.

When encryption of the passwords or authentication strings is enabled, they are encrypted in the CLI regardless of the access level you are using. In the Web management interface, the passwords or authentication strings are encrypted at the read-only access level but are visible at the read-write access level.

The encryption option can be omitted (the default) or can be one of the following.

- **0** – Disables encryption for the password or authentication string you specify with the command. The password or string is shown as clear text in the running-config and the startup-config file. Use this option of you do not want display of the password or string to be encrypted.
- **1** – Assumes that the password or authentication string you enter is the encrypted form, and decrypts the value before using it.

NOTE: If you want the software to assume that the value you enter is the clear-text form, and to encrypt display of that form, do not enter **0** or **1**. Instead, omit the encryption option and allow the software to use the default behavior.

If you specify encryption option **1**, the software assumes that you are entering the encrypted form of the password or authentication string. In this case, the software decrypts the password or string you enter before using the value for authentication. If you accidentally enter option **1** followed by the clear-text version of the password or string, authentication will fail because the value used by the software will not match the value you intended to use.

Change the Timer for OSPF Authentication Changes

When you make an OSPF authentication change, the software uses the authentication-change timer to gracefully implement the change. The software implements the change in the following ways:

- Outgoing OSPF packets – After you make the change, the software continues to use the old authentication to send packets, during the remainder of the current authentication-change interval. After this, the software uses the new authentication for sending packets.
- Inbound OSPF packets – The software accepts packets containing the new authentication and continues to accept packets containing the older authentication for two authentication-change intervals. After the second interval ends, the software accepts packets only if they contain the new authentication key.

The default authentication-change interval is 300 seconds (5 minutes). You change the interval to a value from 0 – 14400 seconds.

OSPF provides graceful authentication change for all the following types of authentication changes in OSPF:

- Changing authentication methods from one of the following to another of the following:
  - Simple text password
  - MD5 authentication
  - No authentication
- Configuring a new simple text password or MD5 authentication key
- Changing an existing simple text password or MD5 authentication key

To change the authentication-change interval, enter a command such as the following at the interface configuration level of the CLI:

```
FastIron(config-if-2/5)#ip ospf auth-change-wait-time 400
```

Syntax: [no] ip ospf auth-change-wait-time <secs>

The <secs> parameter specifies the interval and can be from 0 – 14400 seconds. The default is 300 seconds (5 minutes).

NOTE: For backward compatibility, the **ip ospf md5-authentication key-activation-wait-time** <seconds> command is still supported.

Block Flooding of Outbound LSAs on Specific OSPF Interfaces

By default, the Layer 3 Switch floods all outbound LSAs on all the OSPF interfaces within an area. You can configure a filter to block outbound LSAs on an OSPF interface. This feature is particularly useful when you want to block LSAs from some, but not all, of the interfaces attached to the area.

After you apply filters to block the outbound LSAs, the filtering occurs during the database synchronization and flooding.

If you remove the filters, the blocked LSAs are automatically re-flooded. You do not need to reset OSPF to re-flood the LSAs.

NOTE: You cannot block LSAs on virtual links.

To apply a filter to an OSPF interface to block flooding of outbound LSAs on the interface, enter the following command at the Interface configuration level for that interface.

```
FastIron(config-if-1/1)#ip ospf database-filter all out
```

The command in this example blocks all outbound LSAs on the OSPF interface configured on port 1/1.

Syntax: [no] ip ospf database-filter all out

To remove the filter, enter a command such as the following:

```
FastIron(config-if-1/1)#no ip ospf database-filter all out
```

Configuring an OSPF Non-Broadcast Interface

Platform Support:

- FESX and FSX Layer 3 devices running software release 02.3.01 and later

Layer 3 switches support Non-Broadcast Multi-Access (NBMA) networks. This feature enables you to configure an interface on a Foundry device to send OSPF traffic to its neighbor as unicast packets rather than broadcast packets.

OSPF routers generally use broadcast packets to establish neighbor relationships and broadcast route updates on Ethernet and virtual interfaces (VEs). In this release, as an alternative, you can configure the Foundry device to use unicast packets for this purpose. This can be useful in situations where multicast traffic is not feasible (for example when a firewall does not allow multicast packets).

On a non-broadcast interface, the routers at the other end of this interface must also be configured as non-broadcast and neighbor routers. There is no restriction on the number of routers sharing a non-broadcast interface (for example, through a hub/switch).

NOTE: Only Ethernet interfaces or VEs can be configured as non-broadcast interfaces.

To configure an OSPF interface as a non-broadcast interface, enable the feature on a physical interface or a VE, following the **ip ospf area** statement, and then specify the IP address of the neighbor in the OSPF configuration. The non-broadcast interface configuration must be done on the OSPF routers on both ends of the link.

For example, the following commands configure VE 20 as a non-broadcast interface:

```
FastIron(config)#int ve 20
FastIron(config-vif-20)#ip ospf area 0
FastIron(config-vif-20)#ip ospf network non-broadcast
FastIron(config-vif-20)#exit
```

Syntax: [no] ip ospf network non-broadcast

The following commands specify 1.1.20.1 as an OSPF neighbor address. The address specified must be in the same subnet as a non-broadcast interface.

```
FastIron(config)#router ospf
FastIron(config-ospf-router)#neighbor 1.1.20.1
```

For example, to configure the feature in a network with three routers connected by a hub or switch, each router must have the linking interface configured as a non-broadcast interface, and both of the other routers must be specified as neighbors.

The output of the **show ip ospf interface** command has been enhanced to display information about non-broadcast interfaces and neighbors that are configured in the same subnet. For example:

```
FastIron#show ip ospf interface
v20,OSPF enabled
     IP Address 1.1.20.4, Area 0
     OSPF state BD, Pri 1, Cost 1, Options 2, Type non-broadcast Events 6
     Timers(sec): Transit 1, Retrans 5, Hello 10, Dead 40
     DR:  Router ID 1.1.13.1          Interface Address 1.1.20.5
     BDR: Router ID 2.2.2.1           Interface Address 1.1.20.4
     Neighbor Count = 1, Adjacent Neighbor Count= 2
     Non-broadcast neighbor config: 1.1.20.1, 1.1.20.2, 1.1.20.3, 1.1.20.5,
     Neighbor:         1.1.20.5
     Authentication-Key:None
     MD5 Authentication: Key None, Key-Id None, Auth-change-wait-time 300
```

In the Type field, “non-broadcast” indicates that this is a non-broadcast interface. When the interface type is non-broadcast, the Non-broadcast neighbor config field displays the neighbors that are configured in the same subnet. If no neighbors are configured in the same subnet, a message such as the following is displayed:

```
***Warning! no non-broadcast neighbor config in 1.1.100.1 255.255.255.0
```

Assign Virtual Links

All ABRs (area border routers) must have either a direct or indirect link to the OSPF backbone area (0.0.0.0 or 0). If an ABR does not have a physical link to the area backbone, the ABR can configure a ***virtual link*** to another router within the same area, which has a physical connection to the area backbone.

The path for a virtual link is through an area shared by the neighbor ABR (router with a physical backbone connection), and the ABR requiring a logical connection to the backbone.

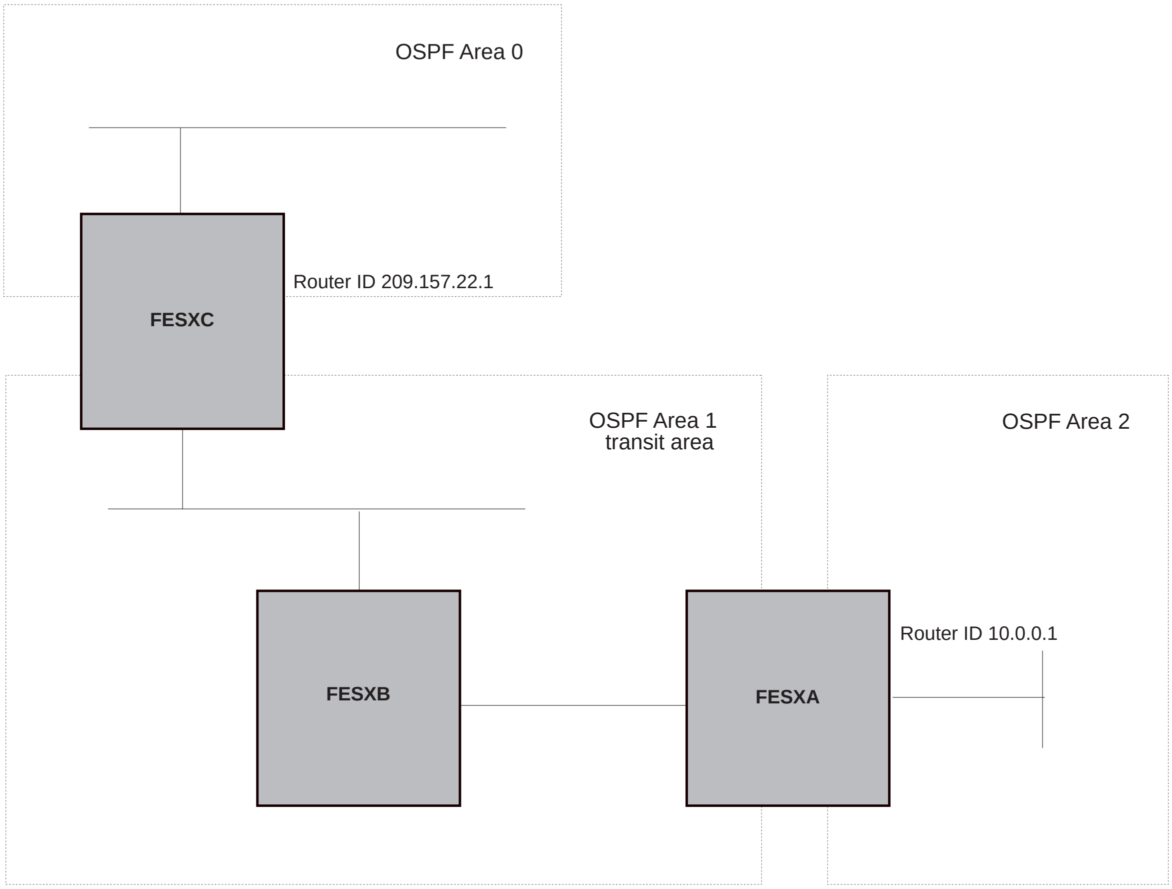
Two parameters fields must be defined for all virtual links—transit area ID and neighbor router.

- The ***transit area ID*** represents the shared area of the two ABRs and serves as the connection point between the two routers. This number should match the area ID value.
- The ***neighbor router*** field is the router ID (IP address) of the router that is physically connected to the backbone, when assigned from the router interface requiring a logical connection. When assigning the parameters from the router with the physical connection, the router ID is the IP address of the router requiring a logical connection to the backbone.

NOTE: By default, the Foundry router ID is the IP address configured on the lowest numbered loopback interface. If the Layer 3 Switch does not have a loopback interface, the default router ID is the lowest numbered IP address configured on the device. For more information or to change the router ID, see “Changing the Router ID” on page 29-24.

NOTE: When you establish an area virtual link, you must configure it on both of the routers (both ends of the virtual link).

Figure 35.6 Defining OSPF Virtual Links within a Network



EXAMPLES:

Figure 35.6 shows an OSPF area border router, FastIronA, that is cut off from the backbone area (area 0). To provide backbone access to FastIronA, you can add a virtual link between FastIronA and FastIronC using area 1 as a transit area. To configure the virtual link, you define the link on the router that is at each end of the link. No configuration for the virtual link is required on the routers in the transit area.

To define the virtual link on FastIronA, enter the following commands:

```
FastIronA(config-ospf-router)#area 1 virtual-link 209.157.22.1
FastIronA(config-ospf-router)#write memory
```

Enter the following commands to configure the virtual link on FastIronC:

```
FastIronC(config-ospf-router)#area 1 virtual-link 10.0.0.1
FastIronC(config-ospf-router)#write memory
```

Syntax: area <ip-addr> | <num> virtual-link <router-id>
[authentication-key | dead-interval | hello-interval | retransmit-interval | transmit-delay <value>]

The **area** <ip-addr> | <num> parameter specifies the transit area.

The <router-id> parameter specifies the router ID of the OSPF router at the remote end of the virtual link. To display the router ID on a Foundry Layer 3 Switch, enter the **show ip** command.

See “Modify Virtual Link Parameters” on page 35-20 for descriptions of the optional parameters.

Modify Virtual Link Parameters

OSPF has some parameters that you can modify for virtual links. Notice that these are the same parameters as the ones you can modify for physical interfaces.

You can modify default values for virtual links using the following CLI command at the ***OSPF router level*** of the CLI, as shown in the following syntax:

Syntax: area <num> | <ip-addr> virtual-link <ip-addr> [authentication-key [0 | 1] <string>] [dead-interval <num>] [hello-interval <num>] [md5-authentication key-activation-wait-time <num> | key-id <num> [0 | 1] key <string>] [retransmit-interval <num>] [transmit-delay <num>]

The parameters are described below. For syntax information, see the *Foundry Switch and Router Command Line Interface Reference*.

Virtual Link Parameter Descriptions

You can modify the following virtual link interface parameters:

Authentication Key: This parameter allows you to assign different authentication methods on a port-by-port basis. OSPF supports three methods of authentication for each interface—none, simple password, and MD5. Only one method of authentication can be active on an interface at a time.

The simple password method of authentication requires you to configure an alphanumeric password on an interface. The password can be up to eight characters long. The simple password setting takes effect immediately. All OSPF packets transmitted on the interface contain this password. All OSPF packets received on the interface are checked for this password. If the password is not present, then the packet is dropped.

The MD5 method of authentication encrypts the authentication key you define. The authentication is included in each OSPF packet transmitted.

MD5 Authentication Key: When simple authentication is enabled, the key is an alphanumeric password of up to eight characters. When MD5 is enabled, the key is an alphanumeric password of up to 16 characters that is later encrypted and included in each OSPF packet transmitted. You must enter a password in this field when the system is configured to operate with either simple or MD5 authentication.

MD5 Authentication Key ID: The Key ID is a number from 1 – 255 and identifies the MD5 key that is being used. This parameter is required to differentiate among multiple keys defined on a router.

MD5 Authentication Wait Time: This parameter determines when a newly configured MD5 authentication key is valid. This parameter provides a graceful transition from one MD5 key to another without disturbing the network. All new packets transmitted after the key activation wait time interval use the newly configured MD5 Key. OSPF packets that contain the old MD5 key are accepted for up to five minutes after the new MD5 key is in operation.

The range for the key activation wait time is from 0 – 14400 seconds. The default value is 300 seconds.

Hello Interval: The length of time between the transmission of hello packets. The range is 1 – 65535 seconds. The default is 10 seconds.

Retransmit Interval: The interval between the re-transmission of link state advertisements to router adjacencies for this interface. The range is 0 – 3600 seconds. The default is 5 seconds.

Transmit Delay: The period of time it takes to transmit Link State Update packets on the interface. The range is 0 – 3600 seconds. The default is 1 second.

Dead Interval: The number of seconds that a neighbor router waits for a hello packet from the current router before declaring the router down. The range is 1 – 65535 seconds. The default is 40 seconds.

Encrypted Display of the Authentication String or MD5 Authentication Key

The optional **0 | 1** parameter with the **authentication-key** and **md5-authentication key-id** parameters affects encryption.

For added security, FastIron devices encrypt display of the password or authentication string. Encryption is enabled by default. The software also provides an optional parameter to disable encryption of a password or authentication string, on an individual OSPF area or OSPF interface basis.

When encryption of the passwords or authentication strings is enabled, they are encrypted in the CLI regardless of the access level you are using. In the Web management interface, the passwords or authentication strings are encrypted at the read-only access level but are visible at the read-write access level.

The encryption option can be omitted (the default) or can be one of the following.

- **0** – Disables encryption for the password or authentication string you specify with the command. The password or string is shown as clear text in the running-config and the startup-config file. Use this option of you do not want display of the password or string to be encrypted.
- **1** – Assumes that the password or authentication string you enter is the encrypted form, and decrypts the value before using it.

NOTE: If you want the software to assume that the value you enter is the clear-text form, and to encrypt display of that form, do not enter **0** or **1**. Instead, omit the encryption option and allow the software to use the default behavior.

If you specify encryption option **1**, the software assumes that you are entering the encrypted form of the password or authentication string. In this case, the software decrypts the password or string you enter before using the value for authentication. If you accidentally enter option **1** followed by the clear-text version of the password or string, authentication will fail because the value used by the software will not match the value you intended to use.

Changing the Reference Bandwidth for the Cost on OSPF Interfaces

Each interface on which OSPF is enabled has a cost associated with it. The Layer 3 Switch advertises its interfaces and their costs to OSPF neighbors. For example, if an interface has an OSPF cost of ten, the Layer 3 Switch advertises the interface with a cost of ten to other OSPF routers.

By default, an interface's OSPF cost is based on the port speed of the interface. The cost is calculated by dividing the reference bandwidth by the port speed. The default reference bandwidth is 100 Mbps, which results in the following default costs:

- 10 Mbps port – 10
- All other port speeds – 1

You can change the reference bandwidth, to change the costs calculated by the software.

The software uses the following formula to calculate the cost:

Cost = reference-bandwidth/interface-speed

If the resulting cost is less than 1, the software rounds the cost up to 1. The default reference bandwidth results in the following costs:

- 10 Mbps port's cost = 100/10 = 10
- 100 Mbps port's cost = 100/100 = 1
- 1000 Mbps port's cost = 100/1000 = 0.10, which is rounded up to 1
- 155 Mbps port's cost = 100/155 = 0.65, which is rounded up to 1
- 622 Mbps port's cost = 100/622 = 0.16, which is rounded up to 1
- 2488 Mbps port's cost = 100/2488 = 0.04, which is rounded up to 1

For 10-Gigabit OSPF interfaces, in order to differentiate the costs between 100 Mbps, 1000 Mbps, and 10,000 Mbps interfaces, you can set the auto-cost reference bandwidth to 10000, whereby each slower link is given a higher cost, as follows:

- 10 Mbps port's cost = 10000/10 = 1000
- 100 Mbps port's cost = 10000/100 = 100
- 1000 Mbps port's cost = 10000/1000 = 10

- 10000 Mbps port’s cost = 10000/10000 = 1

The bandwidth for interfaces that consist of more than one physical port is calculated as follows:

- Trunk group – The combined bandwidth of all the ports.
- Virtual interface – The combined bandwidth of all the ports in the port-based VLAN that contains the virtual interface.

The default reference bandwidth is 100 Mbps. You can change the reference bandwidth to a value from 1 – 4294967.

If a change to the reference bandwidth results in a cost change to an interface, the Layer 3 Switch sends a link-state update to update the costs of interfaces advertised by the Layer 3 Switch.

NOTE: If you specify the cost for an individual interface, the cost you specify overrides the cost calculated by the software.

Interface Types To Which the Reference Bandwidth Does Not Apply

Some interface types are not affected by the reference bandwidth and always have the same cost regardless of the reference bandwidth in use:

- The cost of a loopback interface is always 0.
- The cost of a virtual link is calculated using the Shortest Path First (SPF) algorithm and is not affected by the auto-cost feature.
- The bandwidth for tunnel interfaces is 9 Kbps and is not affected by the auto-cost feature.

Changing the Reference Bandwidth

To change the reference bandwidth, enter a command such as the following at the OSPF configuration level of the CLI:

```
FastIron(config-ospf-router)#auto-cost reference-bandwidth 500
```

The reference bandwidth specified in this example results in the following costs:

- 10 Mbps port’s cost = 500/10 = 50
- 100 Mbps port’s cost = 500/100 = 5
- 1000 Mbps port’s cost = 500/1000 = 0.5, which is rounded up to 1
- 155 Mbps port’s cost = 500/155 = 3.23, which is rounded up to 4
- 622 Mbps port’s cost = 500/622 = 0.80, which is rounded up to 1
- 2488 Mbps port’s cost = 500/2488 = 0.20, which is rounded up to 1

The costs for 10 Mbps, 100 Mbps, and 155 Mbps ports change as a result of the changed reference bandwidth. Costs for higher-speed interfaces remain the same.

Syntax: [no] auto-cost reference-bandwidth <num>

The <num> parameter specifies the reference bandwidth and can be a value from 1 – 4294967. The default is 100. For 10-Gigabit OSPF interfaces, in order to differentiate the costs between 100 Mbps, 1000 Mbps, and 10,000 Mbps interfaces, set the auto-cost reference bandwidth to 10000, whereby each slower link is given a higher cost

To restore the reference bandwidth to its default value and thus restore the default costs of interfaces to their default values, enter the following command:

```
FastIron(config-ospf-router)#no auto-cost reference-bandwidth
```

Define Redistribution Filters

Route redistribution imports and translates different protocol routes into a specified protocol type. On Foundry routers, redistribution is supported for static routes, OSPF, RIP, and BGP4. When you configure redistribution for RIP, you can specify that static, OSPF, or BGP4 routes are imported into RIP routes. Likewise, OSPF redistribution supports the import of static, RIP, and BGP4 routes into OSPF routes. BGP4 supports redistribution of static, RIP, and OSPF routes into BGP4.

NOTE: The Layer 3 Switch advertises the default route into OSPF even if redistribution is not enabled, and even if the default route is learned through an IBGP neighbor. IBGP routes (including the default route) are not redistributed into OSPF by OSPF redistribution (for example, by the OSPF **redistribute** command).

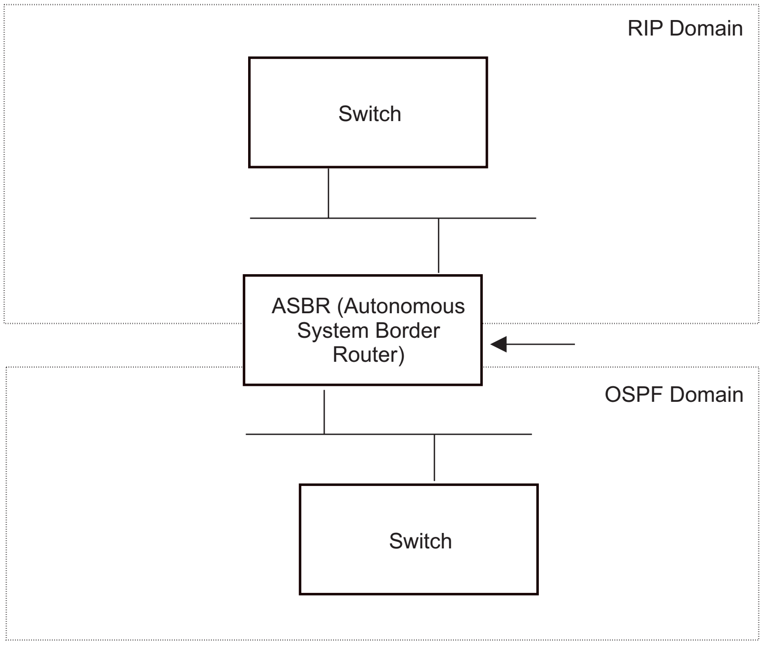
In Figure 35.7 on page 35-23, an administrator wants to configure the FastIron Layer 3 Switch acting as the ASBR (Autonomous System Boundary Router) between the RIP domain and the OSPF domain to redistribute routes between the two domains.

NOTE: The ASBR must be running both RIP and OSPF protocols to support this activity.

To configure for redistribution, define the redistribution tables with deny and permit redistribution filters. Use the **deny** and **permit** redistribute commands for OSPF at the OSPF router level.

NOTE: Do not enable redistribution until you have configured the redistribution filters. If you enable redistribution before you configure the redistribution filters, the filters will not take affect and all routes will be distributed.

Figure 35.7 Redistributing OSPF and Static Routes to RIP Routes



EXAMPLES:

To configure the FastIron Layer 3 Switch acting as an ASBR in Figure 35.7 to redistribute OSPF, BGP4, and static routes into RIP, enter the following commands:

```
FastIronASBR(config)#router rip
FastIronASBR(config-rip-router)#permit redistribute 1 all
FastIronASBR(config-rip-router)#write memory
```

NOTE: Redistribution is permitted for all routes by default, so the **permit redistribute 1 all** command in the example above is shown for clarity but is not required.

You also have the option of specifying import of just OSPF, BGP4, or static routes, as well as specifying that only routes for a specific network or with a specific cost (metric) be imported, as shown in the command syntax below:

Syntax: deny | permit redistribute <filter-num> all | bgp | connected | rip | static
[address <ip-addr> <ip-mask> [match-metric <value> [set-metric <value>]]]

EXAMPLES:

To redistribute RIP, static, and BGP4 routes into OSPF, enter the following commands on the Layer 3 Switch acting as an ASBR:

```
FastIronASBR(config)#router ospf
FastIronASBR(config-ospf-router)#permit redistribute 1 all
FastIronASBR(config-ospf-router)#write memory
```

Syntax: deny | permit redistribute <filter-num> all | bgp | connected | rip | static
address <ip-addr> <ip-mask>
[match-metric <value> | set-metric <value>]

NOTE: Redistribution is permitted for all routes by default, so the **permit redistribute 1 all** command in the example above is shown for clarity but is not required.

You also have the option of specifying import of just OSPF, BGP4, or static routes, as well as specifying that only routes for a specific network or with a specific cost (metric) be imported, as shown in the command syntax below:

Syntax: [no] redistribution bgp | connected | rip | static [route-map <map-name>]

For example, to enable redistribution of RIP and static IP routes into OSPF, enter the following commands.

```
FastIron(config)#router ospf
FastIron(config-ospf-router)#redistribution rip
FastIron(config-ospf-router)#redistribution static
FastIron(config-ospf-router)#write memory
```

NOTE: The **redistribution** command does not perform the same function as the **permit redistribute** and **deny redistribute** commands. The **redistribute** commands allow you to control redistribution of routes by filtering on the IP address and network mask of a route. The **redistribution** commands enable redistribution for routes of specific types (static, directly connected, and so on). Configure all your redistribution filters before enabling redistribution.

NOTE: Do not enable redistribution until you have configured the redistribution filters. If you enable redistribution before you configure the redistribution filters, the filters will not take affect and all routes will be distributed.

Prevent Specific OSPF Routes from Being Installed in the IP Route Table

By default, all OSPF routes in the OSPF route table are eligible for installation in the IP route table. You can configure a distribution list to explicitly deny specific routes from being eligible for installation in the IP route table.

NOTE: This feature does not block receipt of LSAs for the denied routes. The Layer 3 Switch still receives the routes and installs them in the OSPF database. The feature only prevents the software from installing the denied OSPF routes into the IP route table.

To configure an OSPF distribution list:

- Configure a standard or extended ACL that identifies the routes you want to deny. Using a standard ACL lets you deny routes based on the destination network, but does not filter based on the network mask. To also filter based on the destination network's network mask, use an extended ACL.
- Configure an OSPF distribution list that uses the ACL as input.

NOTE: If you change the ACL after you configure the OSPF distribution list, you must clear the IP route table to place the changed ACL into effect. To clear the IP route table, enter the **clear ip route** command at the Privileged EXEC level of the CLI.

The following sections show how to use the CLI to configure an OSPF distribution list. Separate examples are provided for standard and extended ACLs.

NOTE: The examples show named ACLs. However, you also can use a numbered ACL as input to the OSPF distribution list.

Using a Standard ACL as Input to the Distribution List

To use a standard ACL to configure an OSPF distribution list for denying specific routes, enter commands such as the following:

```
FastIron(config)#ip access-list standard no_ip
FastIron(config-std-nacl)#deny 4.0.0.0 0.255.255.255
FastIron(config-std-nacl)#permit any any
FastIron(config-std-nacl)#exit
FastIron(config)#router ospf
FastIron(config-ospf-router)#distribute-list no_ip in
```

The first three commands configure a standard ACL that denies routes to any 4.x.x.x destination network and allows all other routes for eligibility to be installed in the IP route table. The last three commands change the CLI to the OSPF configuration level and configure an OSPF distribution list that uses the ACL as input. The distribution list prevents routes to any 4.x.x.x destination network from entering the IP route table. The distribution list does not prevent the routes from entering the OSPF database.

Syntax: [no] distribute-list <acl-name> | <acl-id> in [<interface type>] [<interface number>]

Syntax: [no] ip access-list standard <acl-name> | <acl-id>

Syntax: deny | permit <source-ip> <wildcard>

The <acl-name> | <acl-id> parameter specifies the ACL name or ID.

The **in** command applies the ACL to incoming route updates.

The <interface number> parameter specifies the interface number on which to apply the ACL. Enter only one valid interface number. If necessary, use the **show interface brief** command to display a list of valid interfaces. If you do not specify an interface, the Foundry device applies the ACL to all incoming route updates.

If you do not specify an interface type and interface number, the device applies the OSPF distribution list to all incoming route updates.

The **deny | permit** parameter indicates whether packets that match the policy are dropped or forwarded.

The <source-ip> parameter specifies the source address for the policy. Since this ACL is input to an OSPF distribution list, the <source-ip> parameter actually is specifying the destination network of the route.

The <wildcard> parameter specifies the portion of the source address to match against. The <wildcard> is a four-part value in dotted-decimal notation (IP address format) consisting of ones and zeros. Zeros in the mask mean the packet's source address must match the <source-ip>. Ones mean any value matches. For example, the <source-ip> and <wildcard> values 4.0.0.0 0.255.255.255 mean that all 4.x.x.x networks match the ACL.

If you want the policy to match on all destination networks, enter **any any**.

If you prefer to specify the wildcard (mask value) in Classless Interdomain Routing (CIDR) format, you can enter a forward slash after the IP address, then enter the number of significant bits in the mask. For example, you can enter the CIDR equivalent of "4.0.0.0 0.255.255.255" as "4.0.0.0/8". The CLI automatically converts the CIDR number into the appropriate ACL mask (where zeros instead of ones are the significant bits) and changes the non-significant portion of the IP address into zeros.

NOTE: If you enable the software to display IP subnet masks in CIDR format, the mask is saved in the file in "/<mask-bits>" format. To enable the software to display the CIDR masks, enter the **ip show-subnet-length** command at the global CONFIG level of the CLI. You can use the CIDR format to configure the ACL entry regardless of whether the software is configured to display the masks in CIDR format.

If you use the CIDR format, the ACL entries appear in this format in the running-config and startup-config files, but are shown with subnet mask in the display produced by the **show ip access-list** command.

Using an Extended ACL as Input to the Distribution List

You can use an extended ACL with an OSPF distribution list to filter OSPF routes based on the network mask of the destination network.

To use an extended ACL to configure an OSPF distribution list for denying specific routes, enter commands such as the following:

```
FastIron(config)#ip access-list extended no_ip
FastIron(config-ext-nacl)#deny ip 4.0.0.0 0.255.255.255 255.255.0.0 0.0.255.255
FastIron(config-ext-nacl)#permit ip any any
FastIron(config-ext-nacl)#exit
FastIron(config)#router ospf
FastIron(config-ospf-router)#distribute-list no_ip in
```

The first three commands configure an extended ACL that denies routes to any 4.x.x.x destination network with a 255.255.0.0 network mask and allows all other routes for eligibility to be installed in the IP route table. The last three commands change the CLI to the OSPF configuration level and configure an OSPF distribution list that uses the ACL as input. The distribution list prevents routes to any 4.x.x.x destination network with network mask 255.255.0.0 from entering the IP route table. The distribution list does not prevent the routes from entering the OSPF database.

Syntax: [no] ip access-list extended <acl-name> | <acl-id>

Syntax: deny | permit <ip-protocol> <source-ip> <wildcard> <destination-ip> <wildcard>

The <acl-name> | <acl-id> parameter specifies the ACL name or ID.

The **deny | permit** parameter indicates whether packets that match the policy are dropped or forwarded.

The <ip-protocol> parameter indicates the type of IP packet you are filtering. When using an extended ACL as input for an OSPF distribution list, specify **ip**.

Since this ACL is input to an OSPF distribution list, the <source-ip> parameter actually specifies the destination network of the route.

The <wildcard> parameter specifies the portion of the source address to match against. The <wildcard> is a four-part value in dotted-decimal notation (IP address format) consisting of ones and zeros. Zeros in the mask mean the packet's source address must match the <source-ip>. Ones mean any value matches. For example, the <source-ip> and <wildcard> values 4.0.0.0 0.255.255.255 mean that all 4.x.x.x networks match the ACL.

If you want the policy to match on all network addresses, enter **any any**.

If you prefer to specify the wildcard (mask value) in Classless Interdomain Routing (CIDR) format, you can enter a forward slash after the IP address, then enter the number of significant bits in the mask. For example, you can enter the CIDR equivalent of “4.0.0.0 0.255.255.255” as “4.0.0.0/8”. The CLI automatically converts the CIDR number into the appropriate ACL mask (where zeros instead of ones are the significant bits) and changes the non-significant portion of the IP address into zeros.

NOTE: If you enable the software to display IP subnet masks in CIDR format, the mask is saved in the file in “/<mask-bits>” format. To enable the software to display the CIDR masks, enter the **ip show-subnet-length** command at the global CONFIG level of the CLI. You can use the CIDR format to configure the ACL entry regardless of whether the software is configured to display the masks in CIDR format.

If you use the CIDR format, the ACL entries appear in this format in the running-config and startup-config files, but are shown with subnet mask in the display produced by the **show ip access-list** commands.

Since this ACL is input to an OSPF distribution list, the <destination-ip> parameter actually specifies the subnet mask of the route.

The <wildcard> parameter specifies the portion of the subnet mask to match against. For example, the <destination-ip> and <wildcard> values 255.255.255.255 0.0.0.255 mean that subnet mask /24 and longer match the ACL.

If you want the policy to match on all network masks, enter **any any**.

Modify Default Metric for Redistribution

The default metric is a global parameter that specifies the cost applied to all OSPF routes by default. The default value is 10. You can assign a cost from 1 – 15.

NOTE: You also can define the cost on individual interfaces. The interface cost overrides the default cost.

To assign a default metric of 4 to all routes imported into OSPF, enter the following commands:

```
FastIron(config)#router ospf
```

```
FastIron(config-ospf-router)#default-metric 4
```

Syntax: default-metric <value>

The <value> can be from 1 – 16,777,215. The default is 10.

Enable Route Redistribution

To enable route redistribution, use one of the following methods.

NOTE: Do not enable redistribution until you have configured the redistribution filters. Otherwise, you might accidentally overload the network with routes you did not intend to redistribute.

To enable redistribution of RIP and static IP routes into OSPF, enter the following commands.

```
FastIron(config)#router ospf
FastIron(config-ospf-router)#redistribution rip
FastIron(config-ospf-router)#redistribution static
FastIron(config-ospf-router)#write memory
```

Example Using a Route Map

To configure a route map and use it for redistribution of routes into OSPF, enter commands such as the following:

```
FastIron(config)#ip route 1.1.0.0 255.255.0.0 207.95.7.30
FastIron(config)#ip route 1.2.0.0 255.255.0.0 207.95.7.30
FastIron(config)#ip route 1.3.0.0 255.255.0.0 207.95.7.30
FastIron(config)#ip route 4.1.0.0 255.255.0.0 207.95.6.30
FastIron(config)#ip route 4.2.0.0 255.255.0.0 207.95.6.30
FastIron(config)#ip route 4.3.0.0 255.255.0.0 207.95.6.30
FastIron(config)#ip route 4.4.0.0 255.255.0.0 207.95.6.30 5
FastIron(config)#route-map abc permit 1
FastIron(config-routemap abc)#match metric 5
FastIron(config-routemap abc)#set metric 8
FastIron(config-routemap abc)#router ospf
FastIron(config-ospf-router)#redistribute static route-map abc
```

The commands in this example configure some static IP routes, then configure a route map and use the route map for redistributing static IP routes into OSPF.

The **ip route** commands configure the static IP routes. The **route-map** command begins configuration of a route map called "abc". The number indicates the route map entry (called the "instance") you are configuring. A route map can contain multiple entries. The software compares packets to the route map entries in ascending numerical order and stops the comparison once a match is found.

The **match** command in the route map matches on routes that have 5 for their metric value (cost). The **set** command changes the metric in routes that match the route map to 8.

The **redistribute static** command enables redistribution of static IP routes into OSPF, and uses route map "abc" to control the routes that are redistributed. In this example, the route map allows a static IP route to be redistributed into OSPF only if the route has a metric of 5, and changes the metric to 8 before placing the route into the OSPF route table.

The following command shows the result of the redistribution filter. Since only one of the static IP routes configured above matches the route map, only one route is redistributed. Notice that the route's metric is 5 before redistribution but is 8 after redistribution.

```
FastIron#show ip ospf database external extensive

Index Aging  LS ID           Router          Netmask  Metric    Flag
1     2      4.4.0.0         10.10.10.60     ffff0000 80000008  0000
```

Syntax: [no] redistribution bgp | connected | rip | static [route-map <map-name>]

The **bgp | connected | rip | static** parameter specifies the route source.

The **route-map** <map-name> parameter specifies the route map name. The following match parameters are valid for OSPF redistribution:

- **match ip address | next-hop** <acl-num>
- **match metric** <num>
- **match tag** <tag-value>

The following set parameters are valid for OSPF redistribution:

- **set ip next hop** <ip-addr>
- **set metric [+ | -]**<num> **| none**
- **set metric-type type-1 | type-2**
- **set tag** <tag-value>

NOTE: You must configure the route map before you configure a redistribution filter that uses the route map.

NOTE: When you use a route map for route redistribution, the software disregards the permit or deny action of the route map.

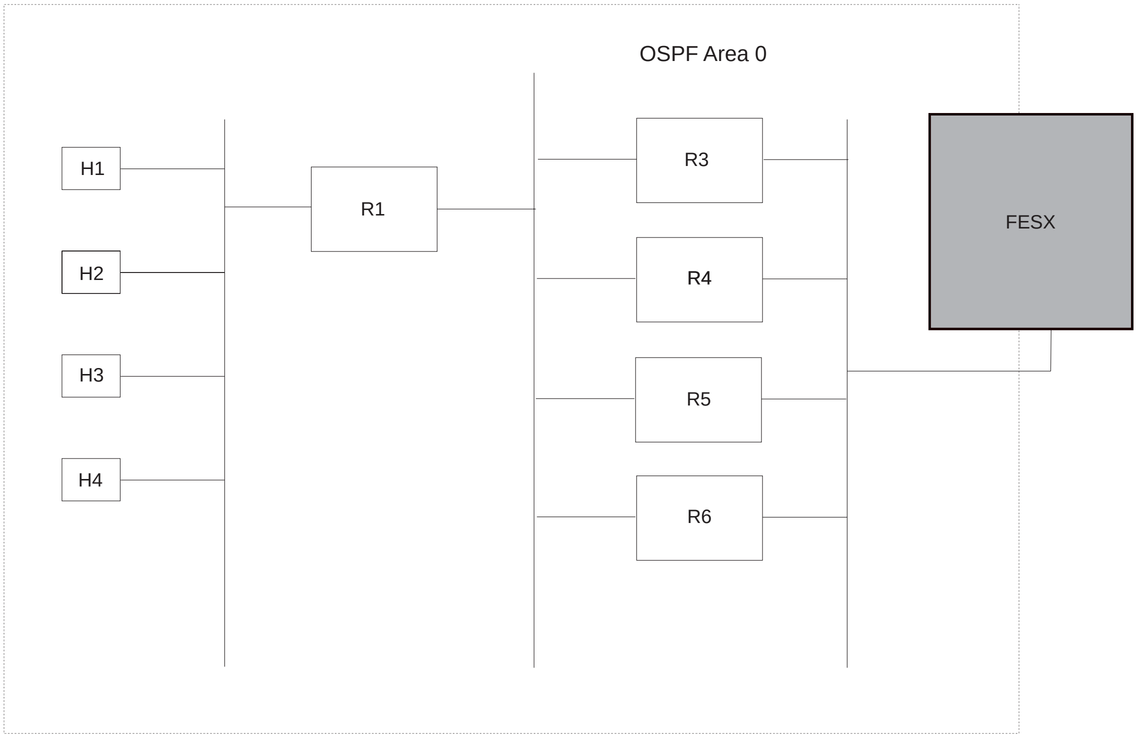
NOTE: For an external route that is redistributed into OSPF through a route map, the metric value of the route remains the same unless the metric is set by a **set metric** command inside the route map. The **default-metric** <num> command has no effect on the route. This behavior is different from a route that is redistributed without using a route map. For a route redistributed without using a route map, the metric is set by the default-metric <num> command.

Disable or Re-Enable Load Sharing

Foundry routers can load share among up to eight equal-cost IP routes to a destination. By default, IP load sharing is enabled. The default is 4 equal-cost paths but you can specify from 2 – 6 paths.

The router software can use the route information it learns through OSPF to determine the paths and costs. Figure 35.8 shows an example of an OSPF network containing multiple paths to a destination (in this case, R1).

Figure 35.8 Example OSPF Network with Four Equal-Cost Paths



In the example in Figure 35.8, the Foundry switch has four paths to R1:

- FI->R3
- FI->R4
- FI->R5
- FI->R6

Normally, the Foundry switch will choose the path to the R1 with the lower metric. For example, if R3's metric is 1400 and R4's metric is 600, the Foundry switch will always choose R4.

However, suppose the metric is the same for all four routers in this example. If the costs are the same, the switch now has four equal-cost paths to R1. To allow the switch to load share among the equal cost routes, enable IP load sharing. The software supports four equal-cost OSPF paths by default when you enable load sharing. You can specify from 2 – 6 paths.

NOTE: The Foundry switch is not source routing in these examples. The switch is concerned only with the paths to the next-hop routers, not the entire paths to the destination hosts.

OSPF load sharing is enabled by default when IP load sharing is enabled. To configure IP load sharing parameters, see "Configuring IP Load Sharing" on page 29-42.

Configure External Route Summarization

When the Layer 3 Switch is an OSPF Autonomous System Boundary Router (ASBR), you can configure it to advertise one external route as an aggregate for all redistributed routes that are covered by a specified address range.

When you configure an address range, the range takes effect immediately. All the imported routes are summarized according to the configured address range. Imported routes that have already been advertised and that fall within the range are flushed out of the AS and a single route corresponding to the range is advertised.

If a route that falls within a configured address range is imported by the Layer 3 Switch, no action is taken if the Layer 3 Switch has already advertised the aggregate route; otherwise the Layer 3 Switch advertises the aggregate route. If an imported route that falls with in a configured address range is removed by the Layer 3 Switch, no action is taken if there are other imported route(s) that fall with in the same address range; otherwise the aggregate route is flushed.

You can configure up to 32 address ranges. The Layer 3 Switch sets the forwarding address of the aggregate route to zero and sets the tag to zero.

If you delete an address range, the advertised aggregate route is flushed and all imported routes that fall within the range are advertised individually.

If an external LSDB overflow condition occurs, all aggregate routes are flushed out of the AS, along with other external routes. When the Layer 3 Switch exits the external LSDB overflow condition, all the imported routes are summarized according to the configured address ranges.

NOTE: If you use redistribution filters in addition to address ranges, the Layer 3 Switch applies the redistribution filters to routes first, then applies them to the address ranges.

NOTE: If you disable redistribution, all the aggregate routes are flushed, along with other imported routes.

To configure a summary address for OSPF routes, enter commands such as the following:

```
FastIron(config-ospf-router)#summary-address 10.1.0.0 255.255.0.0
```

The command in this example configures summary address 10.1.0.0, which includes addresses 10.1.1.0, 10.1.2.0, 10.1.3.0, and so on. For all of these networks, only the address 10.1.0.0 is advertised in external LSAs.

Syntax: summary-address <ip-addr> <ip-mask>

The <ip-addr> parameter specifies the network address.

The <ip-mask> parameter specifies the network mask.

To display the configured summary addresses, enter the following command at any level of the CLI:

```
FastIron#show ip ospf config

OSPF Redistribution Address Ranges currently defined:

Range-Address    Subnetmask
1.0.0.0          255.0.0.0
1.0.1.0          255.255.255.0
1.0.2.0          255.255.255.0
```

Syntax: show ip ospf config

Configure Default Route Origination

When the Layer 3 Switch is an OSPF Autonomous System Boundary Router (ASBR), you can configure it to automatically generate a default external route into an OSPF routing domain. This feature is called "default route origination" or "default information origination".

By default, Foundry Layer 3 Switches do not advertise the default route into the OSPF domain. If you want the Layer 3 Switch to advertise the OSPF default route, you must explicitly enable default route origination.

When you enable OSPF default route origination, the Layer 3 Switch advertises a type 5 default route that is flooded throughout the AS (except stub areas and NSSAs). In addition, internal NSSA ASBRs advertise their default routes as translatable type 7 default routes.

The Layer 3 Switch advertises the default route into OSPF even if OSPF route redistribution is not enabled, and even if the default route is learned through an IBGP neighbor.

NOTE: Foundry Layer 3 Switches never advertise the OSPF default route, regardless of other configuration parameters, unless you explicitly enable default route origination using the following method.

If the Layer 3 Switch is an ASBR, you can use the "always" option when you enable the default route origination. The always option causes the ASBR to create and advertise a default route if it does not already have one configured.

If default route origination is enabled and you disable it, the default route originated by the Layer 3 Switch is flushed. Default routes generated by other OSPF routers are not affected. If you re-enable the feature, the feature takes effect immediately and thus does not require you to reload the software.

NOTE: The ABR (Layer 3 Switch) will not inject the default route into an NSSA by default and the command described in this section will not cause the Layer 3 Switch to inject the default route into the NSSA. To inject the default route into an NSSA, use the **area** <num> | <ip-addr> **nssa default-information-originate** command. See "Assign a Not-So-Stubby Area (NSSA)" on page 35-11.

To enable default route origination, enter the following command:

```
FastIron(config-ospf-router)#default-information-originate
```

To disable the feature, enter the following command:

```
FastIron(config-ospf-router)#no default-information-originate
```

Syntax: [no] default-information-originate [always] [metric <value>] [metric-type <type>]

The **always** parameter advertises the default route regardless of whether the router has a default route. This option is disabled by default.

The **metric** <value> parameter specifies a metric for the default route. If this option is not used, the default metric is used for the route.

The **metric-type** <type> parameter specifies the external link type associated with the default route advertised into the OSPF routing domain. The <type> can be one of the following:

- 1 – Type 1 external route
- 2 – Type 2 external route

If you do not use this option, the default redistribution metric type is used for the route type.

NOTE: If you specify a metric and metric type, the values you specify are used even if you do not use the **always** option.

Modify SPF Timers

The Layer 3 Switch uses the following timers when calculating the shortest path for OSPF routes:

- SPF delay – When the Layer 3 Switch receives a topology change, the software waits before it starts a Shortest Path First (SPF) calculation. By default, the software waits five seconds. You can configure the SPF delay to a value from 0 – 65535 seconds. If you set the SPF delay to 0 seconds, the software immediately begins the SPF calculation after receiving a topology change.
- SPF hold time – The Layer 3 Switch waits for a specific amount of time between consecutive SPF calculations. By default, the Layer 3 Switch waits ten seconds. You can configure the SPF hold time to a value from 0 – 65535 seconds. If you set the SPF hold time to 0 seconds, the software does not wait between consecutive SPF calculations.

You can set the delay and hold time to lower values to cause the Layer 3 Switch to change to alternate paths more quickly in the event of a route failure. Note that lower values require more CPU processing time.

You can change one or both of the timers. To do so, enter commands such as the following:

```
FastIron(config-ospf-router)#timers spf 10 20
```

The command in this example changes the SPF delay to 10 seconds and changes the SPF hold time to 20 seconds.

Syntax: timers spf <delay> <hold-time>

The <delay> parameter specifies the SPF delay.

The <hold-time> parameter specifies the SPF hold time.

To set the timers back to their default values, enter a command such as the following:

```
FastIron(config-ospf-router)#no timers spf 10 20
```

Modify Redistribution Metric Type

The redistribution metric type is used by default for all routes imported into OSPF unless you specify different metrics for individual routes using redistribution filters. Type 2 specifies a big metric (three bytes). Type 1 specifies a small metric (two bytes). The default value is type 2.

To modify the default value to type 1, enter the following command:

```
FastIron(config-ospf-router)#metric-type type1
```

Syntax: metric-type type1 | type2

The default is **type2**.

Modify Administrative Distance

Foundry Layer 3 Switches can learn about networks from various protocols, including Border Gateway Protocol version 4 (BGP4), RIP, and OSPF. Consequently, the routes to a network may differ depending on the protocol from which the routes were learned. The default administrative distance for OSPF routes is 110. See “Changing Administrative Distances” on page 38-28 for a list of the default distances for all route sources.

The router selects one route over another based on the source of the route information. To do so, the router can use the administrative distances assigned to the sources. You can bias the Layer 3 Switch's decision by changing the default administrative distance for RIP routes.

Configuring Administrative Distance Based on Route Type

You can configure a unique administrative distance for each type of OSPF route. For example, you can use this feature to prefer a static route over an OSPF inter-area route but you also want to prefer OSPF intra-area routes to static routes.

The distance you specify influences the choice of routes when the Layer 3 Switch has multiple routes for the same network from different protocols. The Layer 3 Switch prefers the route with the lower administrative distance.

You can specify unique default administrative distances for the following route types:

- Intra-area routes
- Inter-area routes
- External routes

The default for all these OSPF route types is 110.

NOTE: This feature does not influence the choice of routes within OSPF. For example, an OSPF intra-area route is always preferred over an OSPF inter-area route, even if the intra-area route's distance is greater than the inter-area route's distance.

To change the default administrative distances for inter-area routes, intra-area routes, and external routes, enter the following command:

```
FastIron(config-ospf-router)#distance external 100
FastIron(config-ospf-router)#distance inter-area 90
FastIron(config-ospf-router)#distance intra-area 80
```

Syntax: distance external | inter-area | intra-area <distance>

The **external | inter-area | intra-area** parameter specifies the route type for which you are changing the default administrative distance.

The <distance> parameter specifies the new distance for the specified route type. Unless you change the distance for one of the route types using commands such as those shown above, the default is 110.

To reset the administrative distance to its system default (110), enter a command such as the following:

```
FastIron(config-ospf-router)#no distance external 100
```

Configure OSPF Group Link State Advertisement (LSA) Pacing

The Layer 3 Switch paces LSA refreshes by delaying the refreshes for a specified time interval instead of performing a refresh each time an individual LSA's refresh timer expires. The accumulated LSAs constitute a group, which the Layer 3 Switch refreshes and sends out together in one or more packets.

The pacing interval, which is the interval at which the Layer 3 Switch refreshes an accumulated group of LSAs, is configurable to a range from 10 – 1800 seconds (30 minutes). The default is 240 seconds (four minutes). Thus, every four minutes, the Layer 3 Switch refreshes the group of accumulated LSAs and sends the group together in the same packet(s).

Usage Guidelines

The pacing interval is inversely proportional to the number of LSAs the Layer 3 Switch is refreshing and aging. For example, if you have approximately 10,000 LSAs, decreasing the pacing interval enhances performance. If you have a very small database (40 – 100 LSAs), increasing the pacing interval to 10 – 20 minutes might enhance performance slightly.

Changing the LSA Pacing Interval

To change the LSA pacing interval to two minutes (120 seconds), enter the following command:

```
FastIron(config-ospf-router)#timers lsa-group-pacing 120
```

Syntax: [no] timers lsa-group-pacing <secs>

The <secs> parameter specifies the number of seconds and can be from 10 – 1800 (30 minutes). The default is 240 seconds (four minutes).

To restore the pacing interval to its default value, enter the following command:

```
FastIron(config-ospf-router)#no timers lsa-group-pacing
```

Modify OSPF Traps Generated

OSPF traps as defined by RFC 1850 are supported on Foundry routers. OSPF trap generation is enabled on the router, by default.

When using the CLI, you can disable all or specific OSPF trap generation by entering the following CLI command:

```
FastIron(config-ospf-router)#no snmp-server trap ospf
```

To later re-enable the trap feature, enter **snmp-server trap ospf**.

To disable a specific OSPF trap, enter the command as **no snmp-server trap ospf** <ospf-trap>.

These commands are at the OSPF router Level of the CLI.

Here is a summary of OSPF traps supported on Foundry routers, their corresponding CLI commands, and their associated MIB objects from RFC 1850:

- **interface-state-change-trap** – [MIB object: OspfIfstateChange]
- **virtual-interface-state-change-trap** – [MIB object: OspfVirtIfStateChange
- **neighbor-state-change-trap** – [MIB object:ospfNbrStateChange]
- **virtual-neighbor-state-change-trap** – [MIB object: ospfVirtNbrStateChange]
- **interface-config-error-trap** – [MIB object: ospfIfConfigError]
- **virtual-interface-config-error-trap** – [MIB object: ospfVirtIfConfigError]
- **interface-authentication-failure-trap** – [MIB object: ospfIfAuthFailure]
- **virtual-interface-authentication-failure-trap** – [MIB object: ospfVirtIfAuthFailure]
- **interface-receive-bad-packet-trap** – [MIB object: ospfIfrxBadPacket]
- **virtual-interface-receive-bad-packet-trap** – [MIB object: ospfVirtIfRxBadPacket]
- **interface-retransmit-packet-trap** – [MIB object: ospfTxRetransmit]
- **virtual-interface-retransmit-packet-trap** – [MIB object: ospfVirtIfTxRetransmit]
- **originate-lsa-trap** – [MIB object: ospfOriginateLsa]
- **originate-maxage-lsa-trap** – [MIB object: ospfMaxAgeLsa]
- **link-state-database-overflow-trap** – [MIB object: ospfLsdbOverflow]
- **link-state-database-approaching-overflow-trap** – [MIB object: ospfLsdbApproachingOverflow

EXAMPLES:

To stop an OSPF trap from being collected, use the CLI command: **no trap** <ospf-trap>, at the Router OSPF level of the CLI. To disable reporting of the neighbor-state-change-trap, enter the following command:

```
FastIron(config-ospf-router)#no trap neighbor-state-change-trap
```

EXAMPLES:

To reinstate the trap, enter the following command:

```
FastIron(config-ospf-router)#trap neighbor-state-change-trap
```

Syntax: [no] snmp-server trap ospf <ospf-trap>

Modify OSPF Standard Compliance Setting

Foundry routers are configured, by default, to be compliant with the RFC 1583 OSPF V2 specification.

To configure a router to operate with the latest OSPF standard, RFC 2178, enter the following commands:

```
FastIron(config)#router ospf
FastIron(config-ospf-router)#no rfc1583-compatibility
```

Syntax: [no] rfc1583-compatibility

Modify Exit Overflow Interval

If a database overflow condition occurs on a router, the router eliminates the condition by removing entries that originated on the router. The exit overflow interval allows you to set how often a Layer 3 Switch checks to see if the overflow condition has been eliminated. The default value is 0. The range is 0 – 86400 seconds (24 hours). If the configured value of the database overflow interval is zero, then the router never leaves the database overflow condition.

NOTE: FastIron devices dynamically allocate OSPF memory as needed. See “Dynamic OSPF Memory” on page 35-7.

To modify the exit overflow interval to 60 seconds, enter the following command:

```
FastIron(config-ospf-router)#data-base-overflow-interval 60
```

Syntax: database-overflow-interval <value>

The <value> can be from 0 – 86400 seconds. The default is 0 seconds.

Configuring an OSPF Point-to-Point Link

Platform Support:

- FESX devices running software release 02.2.00 and later
- FSX devices running software release 02.3.01 and later

In an OSPF point-to-point link, a direct Layer 3 connection exists between a single pair of OSPF routers, without the need for Designated and Backup Designated routers. In a point-to-point link, neighboring routers become adjacent whenever they can communicate directly. In contrast, in broadcast and non-broadcast multi-access (NBMA) networks, the Designated Router and the Backup Designated Router become adjacent to all other routers attached to the network.

Configuration Notes and Limitations

- This feature is supported on Gigabit Ethernet and 10-Gigabit Ethernet interfaces.
- This feature is supported on physical interfaces. It is not supported on virtual interfaces.
- Foundry supports numbered point-to-point networks, meaning the OSPF router must have an IP interface address which uniquely identifies the router over the network. Foundry does not support unnumbered point-to-point networks.

Configuring an OSPF Point-to-Point Link

To configure an OSPF point-to-point link, enter commands such as the following:

```
FastIron(config)#interface eth 1/5
FastIron(config-if-1/5)#ip ospf network point-to-point
```

This command configures an OSPF point-to-point link on Interface 5 in slot 1.

Syntax: [no] ip ospf network point-to-point

Viewing Configured OSPF Point-to-Point Links

See “Displaying OSPF Neighbor Information” on page 35-41 and “Displaying OSPF Interface Information” on page 35-44.

Specify Types of OSPF Syslog Messages to Log

You can specify which kinds of OSPF-related Syslog messages are logged. By default, the only OSPF messages that are logged are those indicating possible system errors. If you want other kinds of OSPF messages to be logged, you can configure the Foundry device to log them.

NOTE: This feature is not supported on the FGS.

For example, to specify that all OSPF-related Syslog messages be logged, enter the following commands.

```
FastIron(config)#router ospf
FastIron(config-ospf-router)#log all
```

Syntax: [no] log all | adjacency | bad_packet [checksum] | database | memory | retransmit

The **log** command has the following options:

The **all** option causes all OSPF-related Syslog messages to be logged. If you later disable this option with the **no log all** command, the OSPF logging options return to their default settings.

The **adjacency** option logs essential OSPF neighbor state changes, especially on error cases. This option is disabled by default.

The **bad_packet checksum** option logs all OSPF packets that have checksum errors. This option is enabled by default.

The **bad_packet** option logs all other bad OSPF packets. This option is disabled by default.

The **database** option logs OSPF LSA-related information. This option is disabled by default.

The **memory** option logs abnormal OSPF memory usage. This option is enabled by default.

The **retransmit** option logs OSPF retransmission activities. This option is disabled by default.

Clearing OSPF Information

Platform Support:

- FESX and FSX devices running software release 03.0.00 and later

The following kinds of OSPF information can be cleared from a Foundry device’s OSPF link state database and OSPF routing table:

- Routes received from OSPF neighbors. You can clear routes from all OSPF neighbors, or an individual OSPF neighbor, specified either by the neighbor’s IP address or its router ID
- OSPF topology information, including all routes in the OSPF routing table
- All routes in the OSPF routing table that were redistributed from other protocols
- OSPF area information, including routes received from OSPF neighbors within an area, as well as routes imported into the area. You can clear area information for all OSPF areas, or for a specified OSPF area

The OSPF information is cleared dynamically when you enter the command; you do not need to remove statements from the Foundry device’s configuration or reload the software for the change to take effect.

Clearing OSPF Neighbor Information

To clear information on the Foundry device about all OSPF neighbors, enter the following command:

```
FastIron#clear ip ospf neighbor
```

Syntax: clear ip ospf neighbor [ip <ip-addr> | id <ip-addr>] |

This command clears all OSPF neighbors and the OSPF routes exchanged with the neighbors in the Foundry device's OSPF link state database. After this information is cleared, adjacencies with all neighbors are re-established, and routes with these neighbors exchanged again.

To clear information on the Foundry device about OSPF neighbor 10.10.10.1, enter the following command:

```
FastIron#clear ip ospf neighbor ip 10.10.10.1
```

This command clears the OSPF neighbor and the OSPF routes exchanged with neighbor 10.10.10.1 in the Foundry device's OSPF link state database. After this information is cleared, the adjacency with the neighbor is re-established, and routes are exchanged again.

The neighbor router can be specified either by its IP address or its router ID. To specify the neighbor router using its IP address, use the **ip** <ip-addr> parameter. To specify the neighbor router using its router ID, use the **id** <ip-addr> parameter.

Clearing OSPF Topology Information

To clear OSPF topology information on the Foundry device, enter the following command:

```
FastIron#clear ip ospf topology
```

Syntax: clear ip ospf topology

This command clears all OSPF routes from the OSPF routing table, including intra-area, (which includes ABR and ASBR intra-area routes), inter-area, external type 1, external type 2, OSPF default, and OSPF summary routes.

After you enter this command, the OSPF routing table is rebuilt, and valid routes are recomputed from the OSPF link state database. When the OSPF routing table is cleared, OSPF routes in the global routing table are also recalculated. If redistribution is enabled, the routes are imported again.

Clearing Redistributed Routes from the OSPF Routing Table

To clear all routes in the OSPF routing table that were redistributed from other protocols, enter the following command:

```
FastIron#clear ospf redistribution
```

Syntax: clear ospf redistribution

This command clears all routes in the OSPF routing table that are redistributed from other protocols, including direct connected, static, RIP, BGP, and IS-IS. To import redistributed routes from other protocols, use the **redistribution** command at the OSPF configuration level.

Clearing Information for OSPF Areas

To clear information on the Foundry device about all OSPF areas, enter the following command:

```
FastIron#clear ip ospf
```

This command clears all OSPF areas, all OSPF neighbors, and the entire OSPF routing table. After this information has been cleared, adjacencies with all neighbors are re-established, and all OSPF routes are re-learned.

To clear information on the Foundry device about OSPF area 1, enter the following command:

```
FastIron#clear ip ospf area 1
```

This command clears information about the specified area ID. Information about other OSPF areas is not affected. The command clears information about all OSPF neighbors belonging to the specified area, as well as all routes imported into the specified area. Adjacencies with neighbors belonging to the area are re-established, and routes imported into the area are re-learned.

Syntax: clear ip ospf [area <area-id>]

The <area-id> can be specified in decimal format or in IP address format.

Displaying OSPF Information

You can use CLI commands and Web management options to display the following OSPF information:

- Trap, area, and interface information – see "Displaying General OSPF Configuration Information" on page 35-39.
- CPU utilization statistics – see "Displaying CPU Utilization Statistics" on page 35-39.
- Area information – see "Displaying OSPF Area Information" on page 35-41.
- Neighbor information – see "Displaying OSPF Neighbor Information" on page 35-41.
- Interface information – see "Displaying OSPF Interface Information" on page 35-44.
- Route information – see "Displaying OSPF Route Information" on page 35-45.
- External link state information – see "Displaying OSPF External Link State Information" on page 35-47.
- Link state information – see "Displaying OSPF Link State Information" on page 35-48.
- Virtual Neighbor information – see "Displaying OSPF Virtual Neighbor Information" on page 35-49.
- Virtual Link information – see "Displaying OSPF Virtual Link Information" on page 35-49.
- ABR and ASBR information – see "Displaying OSPF ABR and ASBR Information" on page 35-49.
- Trap state information – see "Displaying OSPF Trap Status" on page 35-50.

Displaying General OSPF Configuration Information

To display general OSPF configuration information, enter the following command at any CLI level:

```
FastIron#show ip ospf config
Router OSPF: Enabled
Redistribution: Disabled
Default OSPF Metric: 10
OSPF Redistribution Metric: Type2

OSPF External LSA Limit: 25000

OSPF Database Overflow Interval: 0

RFC 1583 Compatibility: Enabled

Router id: 207.95.11.128

Interface State Change Trap:                     Enabled
Virtual Interface State Change Trap:             Enabled
Neighbor State Change Trap:                      Enabled
Virtual Neighbor State Change Trap:              Enabled
Interface Configuration Error Trap:              Enabled
Virtual Interface Configuration Error Trap:      Enabled
Interface Authentication Failure Trap:           Enabled
Virtual Interface Authentication Failure Trap:   Enabled
Interface Receive Bad Packet Trap:               Enabled
Virtual Interface Receive Bad Packet Trap:       Enabled
Interface Retransmit Packet Trap:                Enabled
Virtual Interface Retransmit Packet Trap:        Enabled
Originate LSA Trap:                              Enabled
Originate MaxAge LSA Trap:                       Enabled
Link State Database Overflow Trap:               Enabled
Link State Database Approaching Overflow Trap:   Enabled

OSPF Area currently defined:
Area-ID          Area-Type Cost
0                normal    0

OSPF Interfaces currently defined:
Ethernet Interface: 3/1-3/2
ip ospf md5-authentication-key-activation-wait-time 300
ip ospf cost 0
ip ospf area 0

Ethernet Interface: v1
ip ospf md5-authentication-key-activation-wait-time 300
ip ospf cost 0
ip ospf area 0
```

Syntax: show ip ospf config

Displaying CPU Utilization Statistics

You can display CPU utilization statistics for OSPF and other IP protocols.

To display CPU utilization statistics for OSPF for the previous one-second, one-minute, five-minute, and fifteen-minute intervals, enter the following command at any level of the CLI:

```
FastIron#show process cpu
Process Name   5Sec(%)   1Min(%)   5Min(%)   15Min(%)   Runtime(ms)
ARP              0.01      0.03      0.09      0.22              9
BGP              0.04      0.06      0.08      0.14             13
GVRP             0.00      0.00      0.00      0.00              0
ICMP             0.00      0.00      0.00      0.00              0
IP               0.00      0.00      0.00      0.00              0
OSPF             0.03      0.06      0.09      0.12             11
RIP              0.00      0.00      0.00      0.00              0
STP              0.00      0.00      0.00      0.00              0
VRRP             0.00      0.00      0.00      0.00              0
```

If the software has been running less than 15 minutes (the maximum interval for utilization statistics), the command indicates how long the software has been running. Here is an example:

```
FastIron#show process cpu
The system has only been up for 6 seconds.
Process Name   5Sec(%)   1Min(%)   5Min(%)   15Min(%)   Runtime(ms)
ARP              0.01      0.00      0.00      0.00              0
BGP              0.00      0.00      0.00      0.00              0
GVRP             0.00      0.00      0.00      0.00              0
ICMP             0.01      0.00      0.00      0.00              1
IP               0.00      0.00      0.00      0.00              0
OSPF             0.00      0.00      0.00      0.00              0
RIP              0.00      0.00      0.00      0.00              0
STP              0.00      0.00      0.00      0.00              0
VRRP             0.00      0.00      0.00      0.00              0
```

To display utilization statistics for a specific number of seconds, enter a command such as the following:

```
FastIron#show process cpu 2
Statistics for last 1 sec and 80 ms
Process Name   Sec(%)   Time(ms)
ARP              0.00       0
BGP              0.00       0
GVRP             0.00       0
ICMP             0.01       1
IP               0.00       0
OSPF             0.00       0
RIP              0.00       0
STP              0.01       0
VRRP             0.00       0
```

When you specify how many seconds' worth of statistics you want to display, the software selects the sample that most closely matches the number of seconds you specified. In this example, statistics are requested for the previous two seconds. The closest sample available is actually for the previous 1 second plus 80 milliseconds.

Syntax: show process cpu [<num>]

The <num> parameter specifies the number of seconds and can be from 1 – 900. If you use this parameter, the command lists the usage statistics only for the specified number of seconds. If you do not use this parameter, the command lists the usage statistics for the previous one-second, one-minute, five-minute, and fifteen-minute intervals.

Displaying OSPF Area Information

To display OSPF area information, enter the following command at any CLI level:

```
FastIron#show ip ospf area

Indx  Area       Type   Cost   SPFR ABR ASBR LSA Chksum(Hex)
1    0.0.0.0     normal  0     1     0     0   1   0000781f
2   192.147.60.0 normal  0     1     0     0   1   0000fee6
3   192.147.80.0 stub    1     1     0     0   2   000181cd
```

Syntax: show ip ospf area [<area-id>] | [<num>]

The <area-id> parameter shows information for the specified area.

The <num> parameter displays the entry that corresponds to the entry number you enter. The entry number identifies the entry's position in the area table.

This display shows the following information.

Table 35.1: CLI Display of OSPF Area Information

This Field...	Displays...
Indx	The row number of the entry in the router's OSPF area table.
Area	The area number.
Type	The area type, which can be one of the following: • nssa • normal • stub
Cost	The area's cost.
SPFR	The SPFR value.
ABR	The ABR number.
ASBR	The ABSR number.
LSA	The LSA number.
Chksum(Hex)	The checksum for the LSA packet. The checksum is based on all the fields in the packet except the age field. The Layer 3 Switch uses the checksum to verify that the packet is not corrupted.

Displaying OSPF Neighbor Information

To display OSPF neighbor information, enter the following command at any CLI level:

```
FastIron#show ip ospf neighbor

Port Address          Pri State      Neigh Address    Neigh ID
8    212.76.7.251     1   full       212.76.7.200     173.35.1.220
```

To display detailed OSPF neighbor information, enter the following command at any CLI level:

```
FastIron#show ip ospf neighbor detail

Port        Address         Pri State      Neigh Address   Neigh ID    Ev Op Cnt
9/1         20.2.0.2        1   FULL/DR    20.2.0.1        2.2.2.2      6  2  0
    Second-to-dead:39
10/1        20.3.0.2        1   FULL/BDR   20.3.0.1        3.3.3.3      5  2  0
    Second-to-dead:36
1/1-1/8     23.5.0.1        1   FULL/DR    23.5.0.2        16.16.16.16  6  2  0
    Second-to-dead:33
2/1-2/2     23.2.0.1        1   FULL/DR    23.2.0.2        15.15.15.15  6  2  0
    Second-to-dead:33
```

Syntax: show ip ospf neighbor [router-id <ip-addr>] | [<num>] | [detail]

The **router-id** <ip-addr> parameter displays only the neighbor entries for the specified router.

The <num> parameter displays only the entry in the specified index position in the neighbor table. For example, if you enter “1”, only the first entry in the table is displayed.

The **detail** parameter displays detailed information about the neighbor routers.

These displays show the following information.

Table 35.2: CLI Display of OSPF Neighbor Information

Field	Description
Port	The port through which the Layer 3 Switch is connected to the neighbor. The port on which an OSPF point-to-point link is configured.
Address	The IP address of this Layer 3 Switch’s interface with the neighbor.
Pri	The OSPF priority of the neighbor. • For multi-access networks, the priority is used during election of the Designated Router (DR) and Backup designated Router (BDR). • For point-to-point links, this field shows one of the following values: • 1 = point-to-point link • 3 = point-to-point link with assigned subnet

Table 35.2: CLI Display of OSPF Neighbor Information (Continued)

Field	Description
State	The state of the conversation between the Layer 3 Switch and the neighbor. This field can have one of the following values: • Down – The initial state of a neighbor conversation. This value indicates that there has been no recent information received from the neighbor. • Attempt – This state is only valid for neighbors attached to non-broadcast networks. It indicates that no recent information has been received from the neighbor. • Init – A Hello packet has recently been seen from the neighbor. However, bidirectional communication has not yet been established with the neighbor. (The router itself did not appear in the neighbor's Hello packet.) All neighbors in this state (or higher) are listed in the Hello packets sent from the associated interface. • 2-Way – Communication between the two routers is bidirectional. This is the most advanced state before beginning adjacency establishment. The Designated Router and Backup Designated Router are selected from the set of neighbors in the 2-Way state or greater. • ExStart – The first step in creating an adjacency between the two neighboring routers. The goal of this step is to decide which router is the master, and to decide upon the initial Database Description (DD) sequence number. Neighbor conversations in this state or greater are called adjacencies. • Exchange – The router is describing its entire link state database by sending Database Description packets to the neighbor. Each Database Description packet has a DD sequence number, and is explicitly acknowledged. Only one Database Description packet can be outstanding at any time. In this state, Link State Request packets can also be sent asking for the neighbor's more recent advertisements. All adjacencies in Exchange state or greater are used by the flooding procedure. In fact, these adjacencies are fully capable of transmitting and receiving all types of OSPF routing protocol packets. • Loading – Link State Request packets are sent to the neighbor asking for the more recent advertisements that have been discovered (but not yet received) in the Exchange state. • Full – The neighboring routers are fully adjacent. These adjacencies will now appear in router links and network link advertisements.
Neigh Address	The IP address of the neighbor. For point-to-point links, the value is as follows: • If the **Pri** field is "1", this value is the IP address of the neighbor router's interface. • If the **Pri** field is "3", this is the subnet IP address of the neighbor router's interface.
Neigh ID	The neighbor router's ID.
Ev	The number of times the neighbor's state changed.
Opt	The sum of the option bits in the Options field of the Hello packet. This information is used by Foundry technical support. See Section A.2 in RFC 2178 for information about the Options field in Hello packets.
Cnt	The number of LSAs that were retransmitted.
Second-to-dead	The amount of time the Foundry device will wait for a HELLO message from each OSPF neighbor before assuming the neighbor is dead.

Displaying OSPF Interface Information

To display OSPF interface information, enter the following command at any CLI level:

```
FastIron#show ip ospf interface 192.168.1.1

Ethernet 2/1,OSPF enabled
     IP Address 192.168.1.1, Area 0
     OSPF state ptr2ptr, Pri 1, Cost 1, Options 2, Type pt-2-pt Events 1
     Timers(sec): Transit 1, Retrans 5, Hello 10, Dead 40
     DR:  Router ID 0.0.0.0           Interface Address 0.0.0.0
     BDR: Router ID 0.0.0.0           Interface Address 0.0.0.0
     Neighbor Count = 0, Adjacent Neighbor Count= 1
     Neighbor: 2.2.2.2
     Authentication-Key:None
     MD5 Authentication: Key None, Key-Id None, Auth-change-wait-time 300
```

Syntax: show ip ospf interface [<ip-addr>]

The <ip-addr> parameter displays the OSPF interface information for the specified IP address.

The following table defines the highlighted fields shown in the above example output of the **show ip ospf interface** command.

Table 35.3: Output of the show ip ospf interface command

This field	Displays
IP Address	The IP address of the interface.
OSPF state	ptr2ptr (point to point)
Pri	The link ID as defined in the router-LSA. This value can be one of the following: 1 = point-to-point link 3 = point-to-point link with an assigned subnet
Cost	The configured output cost for the interface.
Options	OSPF Options (Bit7 - Bit0): • unused:1 • opaque:1 • summary:1 • dont_propagate:1 • nssa:1 • multicast:1 • externals:1 • tos:1

Table 35.3: Output of the show ip ospf interface command

This field	Displays
Type	The area type, which can be one of the following: • Broadcast = 0x01 • NBMA = 0x02 • Point to Point = 0x03 • Virtual Link = 0x04 • Point to Multipoint = 0x05
Events	OSPF Interface Event: • Interface_Up = 0x00 • Wait_Timer = 0x01 • Backup_Seen = 0x02 • Neighbor_Change = 0x03 • Loop_Indication = 0x04 • Unloop_Indication = 0x05 • Interface_Down = 0x06 • Interface_Passive = 0x07
Adjacent Neighbor Count	The number of adjacent neighbor routers.
Neighbor:	The neighbor router’s ID.

Displaying OSPF Route Information

To display OSPF route information for the router, enter the following command at any CLI level:

```
FastIron#show ip ospf routes

Index Destination     Mask            Path_Cost Type2_Cost Path_Type
1     212.95.7.0      255.255.255.0   1         0          Intra
      Adv_Router      Link_State      Dest_Type State      Tag       Flags
      173.35.1.220    212.95.7.251    Network   Valid      00000000  7000
      Paths Out_Port  Next_Hop        Type      Arp_Index  State
      1     5/6       209.95.7.250    OSPF      8          84 00

Index Destination     Mask            Path_Cost Type2_Cost Path_Type
2     11.3.63.0       255.255.255.0   11        0          Inter
      Adv_Router      Link_State      Dest_Type State      Tag       Flags
      209.95.7.250    11.3.63.0       Network   Valid      00000000  0000
      Paths Out_Port  Next_Hop        Type      Arp_Index  State
      1     5/6       209.95.7.250    OSPF      8          84 00
```

Syntax: show ip ospf routes [<ip-addr>]

The <ip-addr> parameter specifies a destination IP address. If you use this parameter, only the route entries for that destination are shown.

This display shows the following information.

Table 35.4: CLI Display of OSPF Route Information

This Field...	Displays...
Index	The row number of the entry in the router’s OSPF route table.
Destination	The IP address of the route's destination.
Mask	The network mask for the route.
Path_Cost	The cost of this route path. (A route can have multiple paths. Each path represents a different exit port for the Layer 3 Switch.)
Type2_Cost	The type 2 cost of this path.
Path_Type	The type of path, which can be one of the following: • Inter – The path to the destination passes into another area. • Intra – The path to the destination is entirely within the local area. • External1 – The path to the destination is a type 1 external route. • External2 – The path to the destination is a type 2 external route.
Adv_Router	The OSPF router that advertised the route to this Foundry Layer 3 Switch.
Link-State	The link state from which the route was calculated.
Dest_Type	The destination type, which can be one of the following: • ABR – Area Border Router • ASBR – Autonomous System Boundary Router • Network – the network
State	The route state, which can be one of the following: • Changed • Invalid • Valid This information is used by Foundry technical support.
Tag	The external route tag.
Flags	State information for the route entry. This information is used by Foundry technical support.
Paths	The number of paths to the destination.
Out_Port	The router port through which the Layer 3 Switch reaches the next hop for this route path.
Next_Hop	The IP address of the next-hop router for this path.
Type	The route type, which can be one of the following: • OSPF • Static Replaced by OSPF
Arp_Index	The index position in the ARP table of the ARP entry for this path's IP address.
State	State information for the path. This information is used by Foundry technical support.

Displaying the Routes that have been Redistributed into OSPF

You can display the routes that have been redistributed into OSPF. To display the redistributed routes, enter the following command at any level of the CLI:

```
FastIron#show ip ospf redistribute route
  4.3.0.0 255.255.0.0 static
  3.1.0.0 255.255.0.0 static
  10.11.61.0 255.255.255.0 connected
  4.1.0.0 255.255.0.0 static
```

In this example, four routes have been redistributed. Three of the routes were redistributed from static IP routes and one route was redistributed from a directly connected IP route.

Syntax: show ip ospf redistribute route [<ip-addr> <ip-mask>]

The <ip-addr> <ip-mask> parameter specifies a network prefix and network mask. Here is an example:

```
FastIron#show ip ospf redistribute route 3.1.0.0 255.255.0.0
  3.1.0.0 255.255.0.0 static
```

Displaying OSPF External Link State Information

To display external link state information, enter the following command at any CLI level:

```
FastIron#show ip ospf database external-link-state

Ospf ext link-state by router ID 130.130.130.241 are in the following:

Area ID         Aging  LS ID           Router          Seq(hex) Chksum   Type
0.0.0.0         279    130.132.75.48   130.130.130.241 80000004 00000ace EXTR
0.0.0.0         278    130.132.88.112  130.130.130.241 80000004 0000f793 EXTR
0.0.0.0         279    130.132.81.208  130.130.130.241 80000004 000081b0 EXTR
0.0.0.0         284    130.132.46.224  130.130.130.241 80000004 000063e1 EXTR
0.0.0.0         285    130.132.40.64   140.140.140.243 80000004 0000ebff EXTR
0.0.0.0         286    130.132.33.160  150.150.150.245 80000004 0000751d EXTR
0.0.0.0         296    130.131.241.16  150.150.150.245 80000004 00002e25 EXTR
```

Syntax: show ip ospf database external-link-state [advertise <num>] | [extensive] | [link-state-id <ip-addr>] | [router-id <ip-addr>] | [sequence-number <num(Hex)>] | [status <num>]

The **advertise** <num> parameter displays the hexadecimal data in the specified LSA packet. The <num> parameter identifies the LSA packet by its position in the router's External LSA table. To determine an LSA packet's position in the table, enter the **show ip ospf external-link-state** command to display the table. See "Displaying the Data in an LSA" on page 35-49 for an example.

The **extensive** option displays the LSAs in decrypted format.

NOTE: You cannot use the **extensive** option in combination with other display options. The entire database is displayed.

The **link-state-id** <ip-addr> parameter displays the External LSAs for the LSA source specified by <IP-addr>.

The **router-id** <ip-addr> parameter shows the External LSAs for the specified OSPF router.

The **sequence-number** <num(Hex)> parameter displays the External LSA entries for the specified hexadecimal LSA sequence number.

The **status** <num> option shows status information.

This display shows the following information.

Table 35.5: CLI Display of OSPF External Link State Information

This Field...	Displays...
Area ID	The OSPF area the router is in.
Aging	The age of the LSA, in seconds.
LS ID	The ID of the link-state advertisement from which the Layer 3 Switch learned this route.
Router	The router IP address.
Seq(hex)	The sequence number of the LSA. The OSPF neighbor that sent the LSA stamps it with a sequence number to enable the Layer 3 Switch and other OSPF routers to determine which LSA for a given route is the most recent.
Chksum	A checksum for the LSA packet, which is based on all the fields in the packet except the age field. The Layer 3 Switch uses the checksum to verify that the packet is not corrupted.
Type	The route type, which is always EXTR (external).

Displaying OSPF Link State Information

To display link state information, enter the following command at any CLI level:

```
FastIron#show ip ospf database link-state
```

Syntax: show ip ospf database link-state [advertise <num>] | [asbr] | [extensive] | [link-state-id <ip-addr>] | [network] | [nssa] | [opaque-area] | [router] | [router-id <ip-addr>] | [sequence-number <num(Hex)>] | [status <num>] | [summary]

The **advertise** <num> parameter displays the hexadecimal data in the specified LSA packet. The <num> parameter identifies the LSA packet by its position in the router's External LSA table. To determine an LSA packet's position in the table, enter the **show ip ospf external-link-state** command to display the table. See "Displaying the Data in an LSA" on page 35-49 for an example.

The **asbr** option shows ASBR information.

The **extensive** option displays the LSAs in decrypted format.

NOTE: You cannot use the **extensive** option in combination with other display options. The entire database is displayed.

The **link-state-id** <ip-addr> parameter displays the External LSAs for the LSA source specified by <IP-addr>.

The **network** option shows network information.

The **nssa** option shows network information.

The **opaque-area** option shows information for opaque areas.

The **router-id** <ip-addr> parameter shows the External LSAs for the specified OSPF router.

The **sequence-number** <num(Hex)> parameter displays the External LSA entries for the specified hexadecimal LSA sequence number.

The **status** <num> option shows status information.

The **summary** option shows summary information.

Displaying the Data in an LSA

You can use the CLI to display the data the Layer 3 Switch received in a specific External LSA packet or other type of LSA packet. For example, to display the LSA data in entry 3 in the External LSA table, enter the following command:

```
FastIron#show ip ospf database external-link-state advertise 3

05 84 02 05 82 83 0d 60 82 82 82 f1 80 00 00 02 e4 05
00 24 ff ff ff f0 80 00 00 0a 00 00 00 00 00 00 00 00
```

Syntax: show ip ospf database external-link-state [advertise <num>] | [link-state-id <ip-addr>] | [router-id <ip-addr>] | [sequence-number <num(Hex)>] | [status <num>]

To determine an external LSA's or other type of LSA's index number, enter one of the following commands to display the appropriate LSA table:

- **show ip ospf database link-state advertise** <num> – This command displays the data in the packet for the specified LSA.
- **show ip ospf database external-link-state advertise** <num> – This command displays the data in the packet for the specified external LSA.

For example, to determine an external LSA's index number, enter the following command:

```
FastIron#show ip ospf external-link-state

Index Aging  LS ID           Router          Seq(hex) Chksum
1     1332   130.132.81.208  130.130.130.241 80000002 000085ae
2     1325   130.132.116.192 130.130.130.241 80000002 0000a37d
3     1330   130.132.88.112  130.130.130.241 80000002 0000fb91
4     1333   130.132.75.48   130.130.130.241 80000002 00000ecc
5     1338   130.132.46.224  130.130.130.241 80000002 000067df
```

additional entries omitted for brevity...

Displaying OSPF Virtual Neighbor Information

To display OSPF virtual neighbor information, enter the following command at any CLI level:

```
FastIron#show ip ospf virtual-neighbor
```

Syntax: show ip ospf virtual-neighbor [<num>]

The <num> parameter displays the table beginning at the specified entry number.

Displaying OSPF Virtual Link Information

To display OSPF virtual link information, enter the following command at any CLI level:

```
FastIron#show ip ospf virtual-link
```

Syntax: show ip ospf virtual-link [<num>]

The <num> parameter displays the table beginning at the specified entry number.

Displaying OSPF ABR and ASBR Information

To display OSPF ABR and ASBR information, enter the following command at any CLI level:

```
FastIron#show ip ospf border-routers
```

Syntax: show ip ospf border-routers [<ip-addr>]

The <ip-addr> parameter displays the ABR and ASBR entries for the specified IP address.

Displaying OSPF Trap Status

All traps are enabled by default when you enable OSPF. To disable or re-enable an OSPF trap, see "Modify OSPF Traps Generated" on page 35-34.

To display the state of each OSPF trap, enter the following command at any CLI level:

```
FastIron#show ip ospf trap

Interface State Change Trap:                      Enabled
Virtual Interface State Change Trap:              Enabled
Neighbor State Change Trap:                       Enabled
Virtual Neighbor State Change Trap:               Enabled
Interface Configuration Error Trap:               Enabled
Virtual Interface Configuration Error Trap:       Enabled
Interface Authentication Failure Trap:            Enabled
Virtual Interface Authentication Failure Trap:    Enabled
Interface Receive Bad Packet Trap:                Enabled
Virtual Interface Receive Bad Packet Trap:        Enabled
Interface Retransmit Packet Trap:                 Enabled
Virtual Interface Retransmit Packet Trap:         Enabled
Originate LSA Trap:                               Enabled
Originate MaxAge LSA Trap:                        Enabled
Link State Database Overflow Trap:                Enabled
Link State Database Approaching Overflow Trap:    Enabled
```

Syntax: show ip ospf trap

Chapter 36
Configuring OSPF Version 3 (IPv6)

This chapter describes how to configure OSPF Version 3 on a Foundry IPv6 Layer 3 Switch.

NOTE: The terms *Layer 3 Switch* and *router* are used interchangeably in this chapter and mean the same thing.

Platform Support:

- FESX and FSX IPv6 devices running software release 04.1.00 and later – L3

Overview

Open Shortest Path First (OSPF) is a link-state routing protocol. OSPF uses link-state advertisements (LSAs) to update neighboring routers about its interfaces and information on those interfaces. The switch floods LSAs to all neighboring routers to update them about the interfaces. Each router maintains an identical database that describes its area topology to help a router determine the shortest path between it and any neighboring router.

This chapter describes the following:

- The differences between OSPF Version 2 (OSPF V2) and OSPF Version 3 (OSPF V3).
- The link state advertisement types for OSPF Version 3.
- How to configure OSPF Version 3.
- How to display OSPF Version 3 information and statistics.

Differences between OSPF V2 and OSPF V3

IPv6 supports OSPF V3 functions similarly to OSPF V2 (the current version that IPv4 supports), except for the following enhancements,:

- Support for IPv6 addresses and prefixes.
- In general, you can configure several IPv6 addresses on a router interface. OSPF V3 imports all or none of the address prefixes configured on a router interface. You cannot select which addresses to import.
- You can run one instance of OSPF Version 2 and one instance of OSPF V3 concurrently on a link.
- Support for IPv6 link state advertisements (LSAs).

In addition, Foundry implements some new commands that are specific to OSPF V3. This chapter describes the commands that are specific to OSPF V3.

NOTE: Although OSPF Versions 2 and 3 function similarly to each other, Foundry has implemented the user interface for each version independently of each other. Therefore, any configuration of OSPF Version 2 features will not affect the configuration of OSPF V3 features and vice versa.

NOTE: You are required to configure a router ID when running only IPv6 routing protocols.

Link State Advertisement Types for OSPF V3

OSPF V3 supports the following types of LSAs:

- Router LSAs (Type 1)
- Network LSAs (Type 2)
- Interarea-prefix LSAs for ABRs (Type 3)
- Interarea-router LSAs for ASBRs (Type 4)
- Autonomous system external LSAs (Type 5)
- Link LSAs (Type 8)
- Intra-area prefix LSAs (Type 9)

For more information about these LSAs, see RFC 2740.

Configuring OSPF V3

To configure OSPF V3, you must do the following:

- Enable OSPF V3 globally.
- Assign OSPF areas.
- Assign router interfaces to an OSPF area.

The following configuration tasks are optional:

- Configure a virtual link between an ABR without a physical connection to a backbone area and the Foundry device in the same area with a physical connection to the backbone area.
- Change the reference bandwidth for the cost on OSPF V3 interfaces.
- Configure the redistribution of routes into OSPF V3.
- Configure default route origination.
- Modify the shortest path first (SPF) timers.
- Modify the administrative distances for OSPF V3 routes.
- Configure the OSPF V3 LSA pacing interval
- Modify how often the Foundry device checks on the elimination of the database overflow condition.
- Modify the external link state database limit.
- Modify the default values of OSPF V3 parameters for router interfaces.
- Disable or re-enable OSPF V3 event logging.

Enabling OSPF V3

Before enabling the Foundry device to run OSPF V3, you must do the following:

- Enable the forwarding of IPv6 traffic on the Foundry device using the **ipv6 unicast-routing** command.
- Enable IPv6 on each interface over which you plan to enable OSPF V3. You enable IPv6 on an interface by configuring an IPv6 address or explicitly enabling IPv6 on that interface.

By default, OSPF V3 is disabled. To enable OSPF V3, you must enable it globally.

To enable OSPF V3 globally, enter the following command:

```
FastIron(config-ospf-router)#ipv6 router ospf
FastIron(config-ospf6-router)#
```

After you enter this command, the Foundry device enters the IPv6 OSPF configuration level, where you can access several commands that allow you to configure OSPF V3.

Syntax: [no] ipv6 router ospf

To disable OSPF V3, enter the **no** form of this command. If you disable OSPF V3, the Foundry device removes all the configuration information for the disabled protocol from the running-config. Moreover, when you save the configuration to the startup-config file after disabling one of these protocols, all the configuration information for the disabled protocol is removed from the startup-config file.

The CLI displays a warning message such as the following:

```
FastIron(config-ospf6-router)# no ipv6 router ospf
ipv6 router ospf mode now disabled. All ospf config data will be lost when writing
to flash!
```

If you have disabled the protocol but have not yet saved the configuration to the startup-config file and reloaded the software, you can restore the configuration information by re-entering the command to enable the protocol (for example, **ipv6 router ospf**). If you have already saved the configuration to the startup-config file and reloaded the software, the information is gone. If you are testing an OSPF configuration and are likely to disable and re-enable the protocol, you might want to make a backup copy of the startup-config file containing the protocol's configuration information. This way, if you remove the configuration information by saving the configuration after disabling the protocol, you can restore the configuration by copying the backup copy of the startup-config file onto the flash memory.

Assigning OSPF V3 Areas

After OSPF V3 is enabled, you can assign OSPF V3 areas. You can assign an IPv4 address or a number as the ***area ID*** for each area. The area ID is representative of all IPv6 addresses (subnets) on a router interface. Each router interface can support one area.

An area can be **normal** or a **stub**.

- Normal – OSPF routers within a normal area can send and receive External Link State Advertisements (LSAs).
- Stub – OSPF routers within a stub area cannot send or receive External LSAs. In addition, OSPF routers in a stub area must use a default route to the area's Area Border Router (ABR) or Autonomous System Boundary Router (ASBR) to send traffic out of the area.

For example, to set up OSPF V3 areas 0.0.0.0, 200.5.0.0, 192.5.1.0, and 195.5.0.0, enter the following commands:

```
FastIron(config-ospf6-router)# area 0.0.0.0
FastIron(config-ospf6-router)# area 200.5.0.0
FastIron(config-ospf6-router)# area 192.5.1.0
FastIron(config-ospf6-router)# area 195.5.0.0
```

Syntax: [no] area <number> | <ipv4-address>

The <number> | <ipv4-address> parameter specifies the area number, which can be a number or in IPv4 address format. If you specify a number, the number can be from 0 – 2,147,483,647.

NOTE: You can assign one area on a router interface.

Assigning a Totally Stubby Area

By default, the Foundry device sends summary LSAs (LSA type 3) into stub areas. You can further reduce the number of LSAs sent into a stub area by configuring the Foundry device to stop sending summary LSAs into the area. You can disable the summary LSAs when you are configuring the stub area or later after you have configured the area.

This feature disables origination of summary LSAs into a stub area, but the Foundry device still accepts summary LSAs from OSPF neighbors and floods them to other areas. The Foundry device can form adjacencies with other routers regardless of whether summarization is enabled or disabled for areas on each router.

When you disable the summary LSAs, the change takes effect immediately. If you apply the option to a previously configured area, the router flushes all of the summary LSAs it has generated (as an ABR) from the area.

NOTE: This feature applies only when the Foundry device is configured as an Area Border Router (ABR) for the area. To completely prevent summary LSAs from being sent to the area, disable the summary LSAs on each OSPF router that is an ABR for the area.

For example, to disable summary LSAs for stub area 40 and specify an additional metric of 99, enter the following command:

```
FastIron(config-ospf6-router)# area 40 stub 99 no-summary
```

Syntax: area <number> | <ipv4-address> stub <metric> [no-summary]

The <number> | <ipv4-address> parameter specifies the area number, which can be a number or in IPv4 address format. If you specify a number, the number can be from 0 – 2,147,483,647.

The **stub** <metric> parameter specifies an additional cost for using a route to or from this area and can be from 1 – 16777215. There is no default. Normal areas do not use the cost parameter.

The **no-summary** parameter applies only to stub areas and disables summary LSAs from being sent into the area.

Assigning Interfaces to an Area

After you define OSPF V3 areas, you must assign router interfaces to the areas. All router interfaces must be assigned to one of the defined areas on an OSPF router. When an interface is assigned to an area, all corresponding subnets on that interface are automatically included in the assignment.

For example, to assign Ethernet interface 3/1 to area 192.5.0.0, enter the following commands:

```
FastIron(config)# interface ethernet 3/1
FastIron(config-if-e100-3/1)# ipv6 ospf area 195.5.0.0
```

Syntax: [no] ipv6 ospf area <number> | <ipv4-address>

The <number> | <ipv4-address> parameter specifies the area number, which can be a number or in IPv4 address format. If you specify a number, the number can be from 0 – 2,147,483,647.

To remove the interface from the specified area, use the **no** form of this command.

Configuring Virtual Links

All ABRs must have either a direct or indirect link to an OSPF backbone area (0.0.0.0 or 0). If an ABR does not have a physical link to a backbone area, you can configure a virtual link from the ABR to another router within the same area that has a physical connection to the backbone area.

The path for a virtual link is through an area shared by the neighbor ABR (router with a physical backbone connection) and the ABR requiring a logical connection to the backbone.

Two parameters must be defined for all virtual links—transit area ID and neighbor router.

- The transit area ID represents the shared area of the two ABRs and serves as the connection point between the two routers. This number should match the area ID value.
- When assigned from the router interface requiring a logical connection, the neighbor router field is the router ID (IPv4 address) of the router that is physically connected to the backbone. When assigned from the router interface with the physical connection, the neighbor router is the router ID (IPv4) address of the router requiring a logical connection to the backbone.

NOTE: By default, the Foundry router ID is the IPv4 address configured on the lowest numbered loopback interface. If the Foundry device does not have a loopback interface, the default router ID is the lowest numbered IPv4 address configured on the device.

NOTE: When you establish an area virtual link, you must configure it on both of the routers (both ends of the virtual link).

For example, imagine that ABR1 in areas 1 and 2 is cut off from the backbone area (area 0). To provide backbone access to ABR1, you can add a virtual link between ABR1 and ABR2 in area 1 using area 1 as a transit area. To configure the virtual link, you define the link on the router that is at each end of the link. No configuration for the virtual link is required on the routers in the transit area.

To define the virtual link on ABR1, enter the following command on ABR1:

```
FastIron(config-ospf6-router)# area 1 virtual-link 209.157.22.1
```

To define the virtual link on ABR2, enter the following command on ABR2:

```
FastIron(config-ospf6-router)# area 1 virtual-link 10.0.0.1
```

Syntax: area <number> | <ipv4-address> virtual-link <router-id>

The **area** <number> | <ipv4-address> parameter specifies the transit area.

The <router-id> parameter specifies the router ID of the OSPF router at the remote end of the virtual link. To display the router ID on a router, enter the **show ip** command.

Assigning a Virtual Link Source Address

When routers at both ends of a virtual link need to communicate with one another, the source address included in the packets must be a global IPv6 address. Therefore, you must determine the global IPv6 address to be used by the routers for communication across the virtual link. You can specify that a router uses the IPv6 global address assigned to one of its interfaces.

For example, to specify the global IPv6 address assigned to Ethernet interface 3/1 on ABR1 as the source address for the virtual link on ABR1, enter the following command on ABR1:

```
FastIron(config-ospf6-router)# virtual-link-if-address interface ethernet 3/1
```

To specify the global IPv6 address assigned to tunnel interface 1 on ABR2 as the source address for the virtual link on ABR2, enter the following command on ABR2:

```
FastIron(config-ospf6-router)# virtual-link-if-address interface tunnel 1
```

Syntax: virtual-link-if-address interface ethernet <port> | loopback <number> | tunnel <number> | ve <number>

The **ethernet | loopback | tunnel | ve** parameter specifies the interface from which the router derives the source IPv6 address for communication across the virtual link. If you specify an Ethernet interface, also specify the port number associated with the interface. If you specify a loopback, tunnel, or VE interface, also specify the number associated with the respective interface.

To delete the source address for the virtual link, use the **no** form of this command.

Modifying Virtual Link Parameters

You can modify the following virtual link parameters:

- Dead-interval: The number of seconds that a neighbor router waits for a hello packet from the current router before declaring the router is down. The range is 1 – 65535 seconds. The default is 40 seconds.
- Hello-interval: The length of time between the transmission of hello packets. The range is 1 – 65535 seconds. The default is 10 seconds.
- Retransmit-interval: The interval between the re-transmission of link state advertisements to router adjacencies for this interface. The range is 0 – 3600 seconds. The default is 5 seconds.
- Transmit-delay: The period of time it takes to transmit Link State Update packets on the interface. The range is 0 – 3600 seconds. The default is 1 second.

NOTE: The values of the **dead-interval** and **hello-interval** parameters must be the same at both ends of a virtual link. Therefore, if you modify the values of these parameters at one end of a virtual link, you must remember to make the same modifications on the other end of the link.

The values of the other virtual link parameters do not require synchronization.

For example, to change the dead interval to 60 seconds on the virtual links defined on ABR1 and ABR2, enter the following command on ABR1:

```
FastIron(config-ospf6-router)# area 1 virtual-link 209.157.22.1
dead-interval 60
```

Enter the following command on ABR2:

```
FastIron(config-ospf6-router)# area 1 virtual-link 10.0.0.1 dead-interval 60
```

Syntax: area <number> | <ipv4-address> virtual-link <router-id> [dead-interval <seconds> | hello-interval <seconds> | retransmit-interval <seconds> | transmit-delay <seconds>]

The **area** <number> | <ipv4-address> parameter specifies the transit area.

The <router-id> parameter specifies the router ID of the OSPF router at the remote end of the virtual link. To display the router ID on a router, enter the **show ip** command.

The **dead-interval**, **hello-interval**, **retransmit-interval**, and **transmit-delay** parameters are discussed earlier in this section.

Changing the Reference Bandwidth for the Cost on OSPF V3 Interfaces

Each interface on which OSPF V3 is enabled has a cost associated with it. The Foundry device advertises its interfaces and their costs to OSPF V3 neighbors. For example, if an interface has an OSPF cost of ten, the Foundry device advertises the interface with a cost of ten to other OSPF routers.

By default, an interface’s OSPF cost is based on the port speed of the interface. The software uses the following formula to calculate the cost:

Cost = reference-bandwidth/interface-speed

By default, the reference bandwidth is 100 Mbps. If the resulting cost is less than 1, the software rounds the cost up to 1. The default reference bandwidth results in the following costs:

- 10 Mbps port’s cost = 100/10 = 10
- 100 Mbps port’s cost = 100/100 = 1
- 1000 Mbps port’s cost = 100/1000 = 0.10, which is rounded up to 1
- 155 Mbps port’s cost = 100/155 = 0.65, which is rounded up to 1
- 622 Mbps port’s cost = 100/622 = 0.16, which is rounded up to 1
- 2488 Mbps port’s cost = 100/2488 = 0.04, which is rounded up to 1

The bandwidth for interfaces that consist of more than one physical port is calculated as follows:

- Trunk group – The combined bandwidth of all the ports.
- Virtual (Ethernet) interface – The combined bandwidth of all the ports in the port-based VLAN that contains the virtual interface.

You can change the default reference bandwidth from 100 Mbps to a value from 1 – 4294967 Mbps.

If a change to the reference bandwidth results in a cost change to an interface, the Foundry device sends a link state update to update the costs of interfaces advertised by the Foundry device.

NOTE: If you specify the cost for an individual interface, the cost you specify overrides the cost calculated by the software.

Some interface types are not affected by the reference bandwidth and always have the same cost regardless of the reference bandwidth in use:

- The cost of a loopback interface is always 0.
- The cost of a virtual link is calculated using the Shortest Path First (SPF) algorithm and is not affected by the auto-cost feature.
- The bandwidth for tunnel interfaces is 9 Kbps and is not affected by the auto-cost feature.

For example, to change the reference bandwidth to 500, enter the following command:

```
FastIron(config-ospf6-router)# auto-cost reference-bandwidth 500
```

The reference bandwidth specified in this example results in the following costs:

- 10 Mbps port’s cost = 500/10 = 50
- 100 Mbps port’s cost = 500/100 = 5
- 1000 Mbps port’s cost = 500/1000 = 0.5, which is rounded up to 1
- 155 Mbps port’s cost = 500/155 = 3.23, which is rounded up to 4
- 622 Mbps port’s cost = 500/622 = 0.80, which is rounded up to 1
- 2488 Mbps port’s cost = 500/2488 = 0.20, which is rounded up to 1

The costs for 10 Mbps, 100 Mbps, and 155 Mbps ports change as a result of the changed reference bandwidth. Costs for higher-speed interfaces remain the same.

Syntax: [no] auto-cost reference-bandwidth <number>

The <number> parameter specifies the reference bandwidth and can be a value from 1 – 4294967. The default is 100.

To restore the reference bandwidth to its default value and thus restore the default costs of interfaces to their default values, enter the **no** form of this command.

Redistributing Routes into OSPF V3

In addition to specifying which routes are redistributed into OSPF V3, you can configure the following aspects related to route redistribution:

- Default metric
- Metric type
- Advertisement of an external aggregate route

Configuring Route Redistribution into OSPF V3

You can configure the Foundry device to redistribute routes from the following sources into OSPF V3:

- IPv6 static routes

- Directly connected IPv6 networks
- RIPng

You can redistribute routes in the following ways:

- By route types, for example, the Foundry device redistributes all IPv6 static and RIPng routes.
- By using a route map to filter which routes to redistribute, for example, the Foundry device redistributes specified IPv6 static and RIPng routes only.

For example, to configure the redistribution of all IPv6 static RIPng routes, enter the following commands:

```
FastIron(config-ospf6-router)# redistribute static
FastIron(config-ospf6-router)# redistribute rip
```

Syntax: [no] redistribute bgp | connected | rip | static [metric <number> | metric-type <type>]

The **connected** | **rip** | **static** keywords specify the route source.

The **metric** <number> parameter specifies the metric used for the redistributed route. If a value is not specified for this option, and the value for the **default-metric** command is set to 0, its default metric, then routes redistributed from the various routing protocols will have the metric value of the protocol from which they are redistributed. For information about the **default-metric** command, see "Modifying Default Metric for Routes Redistributed into OSPF Version 3" on page 36-9

The **metric-type** <type> parameter specifies an OSPF metric type for the redistributed route. You can specify external type 1 or external type 2. If a value is not specified for this option, the Foundry device uses the value specified by the **metric-type** command. For information about modifying the default metric type using the **metric-type** command, see "Modifying Default Metric for Routes Redistributed into OSPF Version 3" on page 36-9

For example, to configure a route map and use it for redistribution of routes into OSPF V3, enter commands such as the following:

```
FastIron(config)# ipv6 route 2001:1::/32 4823:eoff:343e::23
FastIron(config)# ipv6 route 2001:2::/32 4823:eoff:343e::23
FastIron(config)# ipv6 route 2001:3::/32 4823:eoff:343e::23 metric 5
FastIron(config)# route-map abc permit 1
FastIron(config-routemap abc)# match metric 5
FastIron(config-routemap abc)# set metric 8
FastIron(config-routemap abc)# ipv6 router ospf
FastIron(config-ospf6-router)# redistribute static route-map abc
```

The commands in this example configure some static IPv6 routes and a route map, and use the route map for redistributing the static IPv6 routes into OSPF V3.

The **ipv6 route** commands configure the static IPv6 routes. The **route-map** command begins configuration of a route map called "abc". The number indicates the route map entry (called the "instance") you are configuring. A route map can contain multiple entries. The software compares packets to the route map entries in ascending numerical order and stops the comparison once a match is found.

The **match** command in the route map matches on routes that have 5 for their metric value (cost). The **set** command changes the metric in routes that match the route map to 8.

The **redistribute** command configures the redistribution of static IPv6 routes into OSPF V3, and uses route map "abc" to control the routes that are redistributed. In this example, the route map allows a static IPv6 route to be redistributed into OSPF only if the route has a metric of 5, and changes the metric to 8 before placing the route into the OSPF route redistribution table.

Syntax: [no] redistribute bgp | connected | isis | rip | static [route-map <map-name>]

The **bgp** | **connected** | **isis** | **rip** | **static** keywords specify the route source.

The **route-map** <map-name> parameter specifies the route map name. The following match parameters are valid for OSPF V3 redistribution:

- **match ip address** | **next-hop** <acl-number>
- **match metric** <number>
- **match tag** <tag-value>

The following set parameters are valid for OSPF redistribution:

- **set ip next hop** <ipv4-address>
- **set metric** [+ | -] <number> | none
- **set metric-type** type-1 | type-2
- **set tag** <tag-value>

NOTE: You must configure the route map before you configure a redistribution filter that uses the route map.

NOTE: When you use a route map for route redistribution, the software disregards the permit or deny action of the route map.

NOTE: For an external route that is redistributed into OSPF V3 through a route map, the metric value of the route remains the same unless the metric is set by a **set metric** command inside the route map or the **default-metric** <num> command. For a route redistributed without using a route map, the metric is set by the metric parameter if set or the **default-metric** <num> command if the metric parameter is not set.

Modifying Default Metric for Routes Redistributed into OSPF Version 3

The default metric is a global parameter that specifies the cost applied by default to routes redistributed into OSPF V3. The default value is 0.

If the **metric** parameter for the **redistribute** command is not set and the **default-metric** command is set to 0, its default value, then routes redistributed from the various routing protocols will have the metric value of the protocol from which they are redistributed. For information about the **redistribute** command, see “Configuring Route Redistribution into OSPF V3” on page 36-7.

NOTE: You also can define the cost on individual interfaces. The interface cost overrides the default cost. For information about defining the cost on individual interfaces, see “Modifying OSPF V3 Interface Defaults” on page 36-16 and “Changing the Reference Bandwidth for the Cost on OSPF V3 Interfaces” on page 36-6.

To assign a default metric of 4 to all routes imported into OSPF V3, enter the following command:

```
FastIron(config-ospf6-router)# default-metric 4
```

Syntax: [no] default-metric <number>

You can specify a value from 0 – 65535. The default is 0.

To restore the default metric to the default value, use the **no** form of this command.

Modifying Metric Type for Routes Redistributed into OSPF Version 3

The Foundry device uses the **metric-type** parameter by default for all routes redistributed into OSPF V3 unless you specify a different metric type for individual routes using the **redistribute** command. (For more information about using the **redistribute** command, see “Redistributing Routes into OSPF V3” on page 36-7).

A type 1 route specifies a small metric (two bytes), while a type 2 route specifies a big metric (three bytes). The default value is type 2.

To modify the default value of type 2 to type 1, enter the following command:

```
FastIron(config-ospf6-router)# metric-type type1
```

Syntax: [no] metric-type type1 | type2

To restore the metric type to the default value, use the **no** form of this command.

Configuring External Route Summarization

When the Foundry device is an OSPF Autonomous System Boundary Router (ASBR), you can configure it to advertise one external route as an aggregate for all redistributed routes that are covered by a specified IPv6 address range.

When you configure an address range, the range takes effect immediately. All the imported routes are summarized according to the configured address range. Imported routes that have already been advertised and that fall within the range are flushed out of the AS and a single route corresponding to the range is advertised.

If a route that falls within a configured address range is imported by the Foundry device, no action is taken if the device has already advertised the aggregate route; otherwise, the device advertises the aggregate route. If an imported route that falls within a configured address range is removed by the device, no action is taken if there are other imported route(s) that fall with in the same address range; otherwise the aggregate route is flushed.

You can configure up to 32 address ranges. The Foundry device sets the forwarding address of the aggregate route to zero and sets the tag to zero.

If you delete an address range, the advertised aggregate route is flushed and all imported routes that fall within the range are advertised individually.

If an external link state database overflow (LSDB) condition occurs, all aggregate routes are flushed out of the AS, along with other external routes. When the Foundry device exits the external LSDB overflow condition, all the imported routes are summarized according to the configured address ranges.

NOTE: If you use redistribution filters in addition to address ranges, the Foundry device applies the redistribution filters to routes first, then applies them to the address ranges.

NOTE: If you disable redistribution, all the aggregate routes are flushed, along with other imported routes.

NOTE: This option affects only imported, type 5 external routes. A single type 5 LSA is generated and flooded throughout the AS for multiple external routes.

To configure the summary address 2201::/24 for routes redistributed into OSPF V3, enter the following command:

```
FastIron(config-ospf6-router)# summary-address 2201::/24
```

In this example, the summary prefix 2201::/24 includes addresses 2201::/1 through 2201::/24. Only the address FEC0::/24 is advertised in an external link-state advertisement.

Syntax: summary-address <ipv6-prefix>/<prefix-length>

You must specify the <ipv6-prefix> parameter in hexadecimal using 16-bit values between colons as documented in RFC 2373.

You must specify the <prefix-length> parameter as a decimal value. A slash mark (/) must follow the <ipv6-prefix> parameter and precede the <prefix-length> parameter.

Filtering OSPF V3 Routes

You can filter the routes to be placed in the OSPF V3 route table by configuring distribution lists. OSPF V3 distribution lists can be applied globally or to an interface.

The functionality of OSPF V3 distribution lists is similar to that of OSPFv2 distribution lists. However, unlike OSPFv2 distribution lists, which filter routes based on criteria specified in an Access Control List (ACL), OSPF V3 distribution lists can filter routes using information specified in an IPv6 prefix list or a route map.

Configuration Examples

The following sections show examples of filtering OSPF V3 routes using prefix lists globally and for a specific interface, as well as filtering OSPF V3 routes using a route map.

You can configure the device to use all three types of filtering. When you do this, filtering using route maps has higher priority over filtering using global prefix lists. Filtering using prefix lists for a specific interface has lower priority than the other two filtering methods.

The example in this section assume the following routes are in the OSPF V3 route table:

```
FastIron# show ipv6 ospf route

  Current Route count: 5
    Intra: 3 Inter: 0 External: 2 (Type1 0/Type2 2)
    Equal-cost multi-path: 0
    Destination                    Options   Area             Cost Type2 Cost
    Next Hop Router                Outgoing Interface
*IA 3001::/64                      --------- 0.0.0.1             0 0
    ::                             ve 10
*E2 3010::/64                      --------- 0.0.0.0            10 0
    fe80::2e0:52ff:fe00:10         ve 10
*IA 3015::/64                      V6E---R-- 0.0.0.0            11 0
    fe80::2e0:52ff:fe00:10         ve 10
*IA 3020::/64                      --------- 0.0.0.0            10 0
    ::                             ve 11
*E2 6001:5000::/64                 --------- 0.0.0.0            10 0
    fe80::2e0:52ff:fe00:10         ve 10
```

Configuring an OSPF V3 Distribution List Using an IPv6 Prefix List as Input

The following example illustrates how to use an IPv6 prefix list is used to filter OSPF V3 routes.

To specify an IPv6 prefix list called filterOspfRoutes that denies route 3010::/64, enter the following commands:

```
FastIron(config)# ipv6 prefix-list  filterOspfRoutes seq 5 deny 3010::/64
FastIron(config)# ipv6 prefix-list  filterOspfRoutes seq 7 permit ::/0 ge 1 le 128
```

Syntax: ipv6 prefix-list <name> [seq <seq-value>] [description <string>] deny | permit <ipv6-addr>/<mask-bits> [ge <ge-value>] [le <le-value>]

To configure a distribution list that applies the filterOspfRoutes prefix list globally:

```
FastIron(config)# ipv6 router ospf
FastIron(config-ospf6-router)# distribute-list prefix-list filterOspfRoutes in
```

Syntax: [no] distribute-list prefix-list <name> in [<interface>]

After this distribution list is configured, route 3010::/64 would be omitted from the OSPF V3 route table:

```
FastIron# show ipv6 ospf route

  Current Route count: 4
    Intra: 3 Inter: 0 External: 1 (Type1 0/Type2 1)
    Equal-cost multi-path: 0
    Destination                    Options   Area              Cost Type2 Cost
    Next Hop Router                Outgoing Interface
*IA 3001::/64                      --------- 0.0.0.1              0 0
    ::                             ve 10
*IA 3015::/64                      V6E---R-- 0.0.0.0             11 0
    fe80::2e0:52ff:fe00:10         ve 10
*IA 3020::/64                      --------- 0.0.0.0             10 0
    ::                             ve 11
*E2 6001:5000::/64                 --------- 0.0.0.0             10 0
    fe80::2e0:52ff:fe00:10         ve 10
```

The following commands specify an IPv6 prefix list called filterOspfRoutesVe that denies route 3015::/64:

```
FastIron(config)# ipv6 prefix-list filterOspfRoutesVe seq 5 deny 3015::/64
FastIron(config)# ipv6 prefix-list filterOspfRoutesVe seq 10 permit ::/0 ge 1 le 128
```

The following commands configure a distribution list that applies the filterOspfRoutesVe prefix list to routes pointing to virtual interface 10:

```
FastIron(config)# ipv6 router ospf
FastIron(config-ospf6-router)# distribute-list prefix-list filterOspfRoutes in ve 10
```

After this distribution list is configured, route 3015::/64, pointing to virtual interface 10, would be omitted from the OSPF V3 route table:

```
FastIron# show ipv6 ospf route

  Current Route count: 4
    Intra: 3 Inter: 0 External: 1 (Type1 0/Type2 1)
    Equal-cost multi-path: 0
    Destination                    Options   Area              Cost Type2 Cost
    Next Hop Router                Outgoing Interface
*IA 3001::/64                      --------- 0.0.0.1              0 0
    ::                             ve 10
*E2 3010::/64                      --------- 0.0.0.0             10 0
    fe80::2e0:52ff:fe00:10         ve 10
*IA 3020::/64                      --------- 0.0.0.0             10 0
    ::                             ve 11
*E2 6001:5000::/64                 --------- 0.0.0.0             10 0
    fe80::2e0:52ff:fe00:10         ve 10
```

Configuring an OSPF V3 Distribution List Using a Route Map as Input

The following commands configure a route map that matches internal routes:

```
FastIron(config)# route-map allowInternalRoutes permit 10
FastIron(config-routemap allowInternalRoutes)# match route-type internal
```

See “Policy-Based Routing” in the *Foundry FastIron Configuration Guide* for information on configuring route maps.

The following commands configure a distribution list that applies the allowInternalRoutes route map globally to OSPF V3 routes:

```
FastIron(config)# ipv6 router ospf
FastIron(config-ospf6-router)# distribute-list route-map allowinternalroutes in
```

Syntax: [no] distribute-list route-map <name> in

After this distribution list is configured, the internal routes would be included, and the external routes would be omitted from the OSPF V3 route table:

```
FastIron# show ipv6 ospf route

  Current Route count: 3
    Intra: 3 Inter: 0 External: 0 (Type1 0/Type2 0)
    Equal-cost multi-path: 0
    Destination                    Options   Area             Cost Type2 Cost
    Next Hop Router                Outgoing Interface
*IA 3001::/64                      --------- 0.0.0.1             0 0
    ::                             ve 10
*IA 3015::/64                      V6E---R-- 0.0.0.0            11 0
    fe80::2e0:52ff:fe00:10         ve 10
*IA 3020::/64                      --------- 0.0.0.0            10 0
    ::                             ve 11
```

Configuring Default Route Origination

When the Foundry device is an OSPF Autonomous System Boundary Router (ASBR), you can configure it to automatically generate a default external route into an OSPF V3 routing domain. This feature is called “default route origination” or “default information origination.”

By default, the Foundry device does not advertise the default route into the OSPF V3 domain. If you want the device to advertise the OSPF default route, you must explicitly enable default route origination.

When you enable OSPF default route origination, the device advertises a type 5 default route that is flooded throughout the AS (except stub areas).

The device advertises the default route into OSPF even if OSPF route redistribution is not enabled, and even if the default route is learned through an IBGP neighbor.

NOTE: The Foundry device does not advertise the OSPF default route, regardless of other configuration parameters, unless you explicitly enable default route origination.

If default route origination is enabled and you disable it, the default route originated by the device is flushed. Default routes generated by other OSPF routers are not affected. If you re-enable the feature, the feature takes effect immediately and thus does not require you to reload the software.

For example, to create and advertise a default route with a metric of 2 and as a type 1 external route, enter the following command:

```
FastIron(config-ospf6-router)# default-information-originate always metric 2
metric-type type1
```

Syntax: [no] default-information-originate [always] [metric <value>] [metric-type <type>]

The **always** keyword originates a default route regardless of whether the device has learned a default route. This option is disabled by default.

The **metric** <value> parameter specifies a metric for the default route. If this option is not used, the value of the **default-metric** command is used for the route. For information about this command, see “Modifying Default Metric for Routes Redistributed into OSPF Version 3” on page 36-9

The **metric-type** <type> parameter specifies the external link type associated with the default route advertised into the OSPF routing domain. The <type> can be one of the following:

- 1 – Type 1 external route
- 2 – Type 2 external route

If you do not use this option, the default redistribution metric type is used for the route type.

NOTE: If you specify a metric and metric type, the values you specify are used even if you do not use the always option.

To disable default route origination, enter the **no** form of the command.

Modifying Shortest Path First Timers

The Foundry device uses the following timers when calculating the shortest path for OSPF V3 routes:

- SPF delay – When the Foundry device receives a topology change, the software waits before it starts a Shortest Path First (SPF) calculation. By default, the software waits 5 seconds. You can configure the SPF delay to a value from 0 – 65535 seconds. If you set the SPF delay to 0 seconds, the software immediately begins the SPF calculation after receiving a topology change.
- SPF hold time – The Foundry device waits a specific amount of time between consecutive SPF calculations. By default, the device waits 10 seconds. You can configure the SPF hold time to a value from 0 – 65535 seconds. If you set the SPF hold time to 0 seconds, the software does not wait between consecutive SPF calculations.

You can set the SPF delay and hold time to lower values to cause the device to change to alternate paths more quickly if a route fails. Note that lower values for these parameters require more CPU processing time.

You can change one or both of the timers.

NOTE: If you want to change only one of the timers, for example, the SPF delay timer, you must specify the new value for this timer as well as the current value of the SPF hold timer, which you want to retain. The Foundry device does not accept only one timer value.

To change the SPF delay to 10 seconds and the SPF hold to 20 seconds, enter the following command:

```
FastIron(config-ospf6-router)# timers spf 10 20
```

Syntax: timers spf <delay> <hold-time>

For the <delay> and <hold-time> parameters, specify a value from 0 – 65535 seconds.

To set the timers back to their default values, enter the **no** version of this command.

Modifying Administrative Distance

The Foundry device can learn about networks from various protocols, including IPv6, RIPng, and OSPF V3. Consequently, the routes to a network may differ depending on the protocol from which the routes were learned. By default, the administrative distance for OSPF V3 routes is 110.

The device selects one route over another based on the source of the route information. To do so, the device can use the administrative distances assigned to the sources. You can influence the device’s decision by changing the default administrative distance for OSPF V3 routes.

Configuring Administrative Distance Based on Route Type

You can configure a unique administrative distance for each type of OSPF V3 route. For example, you can use this feature to influence the Foundry device to prefer a static route over an OSPF inter-area route and to prefer OSPF intra-area routes to static routes.

The distance you specify influences the choice of routes when the device has multiple routes to the same network from different protocols. The device prefers the route with the lower administrative distance.

You can specify unique default administrative distances for the following OSPF V3 route types:

- Intra-area routes
- Inter-area routes
- External routes

The default for all of these OSPF V3 route types is 110.

NOTE: This feature does not influence the choice of routes within OSPF V3. For example, an OSPF intra-area route is always preferred over an OSPF inter-area route, even if the intra-area route's distance is greater than the inter-area route's distance.

For example, to change the default administrative distances for intra-area routes to 80, inter-area routes to 90, and external routes to 100, enter the following commands:

```
FastIron(config-ospf6-router)# distance intra-area 80
FastIron(config-ospf6-router)# distance inter-area 90
FastIron(config-ospf6-router)# distance external 100
```

Syntax: distance external | inter-area | intra-area <distance>

The **external** | **inter-area** | **intra-area** keywords specify the route type for which you are changing the default administrative distance.

The <distance> parameter specifies the new distance for the specified route type. You can specify a value from 1 – 255.

To reset the administrative distance of a route type to its system default, enter the **no** form of this command.

Configuring the OSPF V3 LSA Pacing Interval

The Foundry device paces OSPF V3 LSA refreshes by delaying the refreshes for a specified time interval instead of performing a refresh each time an individual LSA's refresh timer expires. The accumulated LSAs constitute a group, which the Foundry device refreshes and sends out together in one or more packets.

The pacing interval, which is the interval at which the Foundry device refreshes an accumulated group of LSAs, is configurable to a range from 10 – 1800 seconds (30 minutes). The default is 240 seconds (four minutes). Thus, every four minutes, the Foundry device refreshes the group of accumulated LSAs and sends the group together in the same packet(s).

The pacing interval is inversely proportional to the number of LSAs the Foundry device is refreshing and aging. For example, if you have approximately 10,000 LSAs, decreasing the pacing interval enhances performance. If you have a very small database (40 – 100 LSAs), increasing the pacing interval to 10 – 20 minutes might enhance performance only slightly.

To change the OSPF V3 LSA pacing interval to two minutes (120 seconds), enter the following command:

```
FastIron(config)# ipv6 router ospf
FastIron(config-ospf6-router)# timers lsa-group-pacing 120
```

Syntax: [no] timers lsa-group-pacing <seconds>

The <seconds> parameter specifies the number of seconds and can be from 10 – 1800 (30 minutes). The default is 240 seconds (four minutes).

To restore the pacing interval to its default value, use the **no** form of the command.

Modifying Exit Overflow Interval

If a database overflow condition occurs on the Foundry device, the device eliminates the condition by removing entries that originated on the device. The exit overflow interval allows you to set how often a device checks to see if the overflow condition has been eliminated. The default value is 0. If the configured value of the database overflow interval is 0, then the device never leaves the database overflow condition.

For example, to modify the exit overflow interval to 60 seconds, enter the following command:

```
FastIron(config-ospf6-router)# database-overflow-interval 60
```

Syntax: [no] auto-cost reference-bandwidth <number>

The <seconds> parameter can be a value from 0 – 86400 seconds (24 hours).

To reset the exit overflow interval to its system default, enter the **no** form of this command.

Modifying External Link State Database Limit

By default, the link state database can hold a maximum of 2000 entries for external (type 5) LSAs. You can change the maximum number of entries from 500 – 8000. After changing this limit, make sure to save the running-config file and reload the software. The change does not take effect until you reload or reboot the software.

For example, to change the maximum number entries from the default of 2000 to 3000, enter the following command:

```
FastIron(config-ospf6-router)# external-lsdb-limit 3000
```

Syntax: ipv6 ospf area <number> | <ipv4-address>

The <entries> parameter can be a numerical value from 500 – 8000 seconds.

To reset the maximum number of entries to its system default, enter the **no** form of this command.

Modifying OSPF V3 Interface Defaults

OSPF V3 has interface parameters that you can configure. For simplicity, each of these parameters has a default value. No change to these default values is required except as needed for specific network configurations.

You can modify the default values for the following OSPF interface parameters:

- Cost: Indicates the overhead required to send a packet across an interface. You can modify the cost to differentiate between 100 Mbps and 1000 Mbps (1 Gbps) links. The command syntax is **ipv6 ospf cost** <number>. The default cost is calculated by dividing 100 million by the bandwidth. For 10 Mbps links, the cost is 10. The cost for both 100 Mbps and 1000 Mbps links is 1, because the speed of 1000 Mbps was not in use at the time the OSPF cost formula was devised.
- Dead-interval: Indicates the number of seconds that a neighbor router waits for a hello packet from the current router before declaring the router down. The command syntax is **ipv6 ospf dead-interval** <seconds>. The value can be from 1 – 2147483647 seconds. The default is 40 seconds.
- Hello-interval: Represents the length of time between the transmission of hello packets. The command syntax is **ipv6 ospf hello-interval** <seconds>. The value can be from 1 – 65535 seconds. The default is 10 seconds.
- Instance: Indicates the number of OSPF V3 instances running on an interface. The command syntax is **ipv6 ospf instance** <number>. The value can be from 0 – 255. The default is 1.
- MTU-ignore: Allows you to disable a check that verifies the same MTU is used on an interface shared by neighbors. The command syntax is **ipv6 ospf mtu-ignore**. By default, the mismatch detection is enabled.
- Network: Allows you to configure the OSPF network type. The command syntax is **ipv6 ospf network** [**point-to-multipoint**]. The default setting of the parameter depends on the network type.
- Passive: When you configure an OSPF interface to be passive, that interface does not send or receive OSPF route updates. This option affects all IPv6 subnets configured on the interface. The command syntax is **ipv6 ospf passive**. By default, all OSPF interfaces are active and thus can send and receive OSPF route information. Since a passive interface does not send or receive route information, the interface is in effect a stub network.
- Priority: Allows you to modify the priority of an OSPF router. The priority is used when selecting the designated router (DR) and backup designated routers (BDRs). The command syntax is **ipv6 ospf priority** <number>. The value can be from 0 – 255. The default is 1. If you set the priority to 0, the router does not participate in DR and BDR election.

- Retransmit-interval: The time between retransmissions of LSAs to adjacent routers for an interface. The command syntax is **ipv6 ospf retransmit-interval** <seconds>. The value can be from 0 – 3600 seconds. The default is 5 seconds.
- Transmit-delay: The time it takes to transmit Link State Update packets on this interface. The command syntax is **ipv6 ospf transmit-delay** <seconds>. The value can be from 0 – 3600 seconds. The default is 1 second.

Disabling or Re-enabling Event Logging

OSPF V3 does not currently support the generation of SNMP traps. Instead, you can disable or re-enable the logging of OSPF V3-related events such as neighbor state changes and database overflow conditions. By default, the Foundry device logs these events.

To disable the logging of events, enter the following command:

```
FastIron(config-ospf6-router)# no log-status-change
```

Syntax: [no] log-status-change

To re-enable the logging of events, enter the following command:

```
FastIron(config-ospf6-router)# log-status-change
```

Displaying OSPF V3 Information

You can display the information for the following OSPF V3 parameters:

- Areas
- Link state databases
- Interfaces
- Memory usage
- Neighbors
- Redistributed routes
- Routes
- SPF
- Virtual links
- Virtual neighbors

Displaying OSPF V3 Area Information

To display global OSPF V3 area information for the Foundry device, enter the following command at any CLI level:

```
FastIron# show ipv6 ospf area
Area 0:
  Interface attached to this area: loopback 2 ethe 3/2 tunnel 2
  Number of Area scoped LSAs is 6
  Statistics of Area 0:
    SPF algorithm executed 16 times
    SPF last updated: 335256 sec ago
    Current SPF node count: 3
      Router: 2 Network: 1
      Maximum of Hop count to nodes: 2
...
```

Syntax: show ipv6 ospf area [<area-id>]

You can specify the <area-id> parameter in the following formats:

- As an IPv4 address, for example, 192.168.1.1.
- As a numerical value from 0 – 2,147,483,647.

The <area-id> parameter restricts the display to the specified OSPF area.

This display shows the following information:

Table 36.1:OSPF V3 area information fields

This Field...	Displays...
Area	The area number.
Interface attached to this area	The router interfaces attached to the area.
Number of Area scoped LSAs	Number of LSAs with a scope of the specified area.
SPF algorithm executed	The number of times the OSPF Shortest Path First (SPF) algorithm is executed within the area.
SPF last updated	The interval in seconds that the SPF algorithm was last executed within the area.
Current SPF node count	The current number of SPF nodes in the area.
Router	Number of router LSAs in the area.
Network	Number of network LSAs in the area.
Indx	The row number of the entry in the router's OSPF area table.
Area	The area number.
Maximum hop count to nodes.	The maximum number of hop counts to an SPF node within the area.

Displaying OSPF V3 Database Information

You can display a summary of the Foundry device's link state database or detailed information about a specified LSA type.

To display a summary of a device's link state database, enter the following command at any CLI level:

```
FastIron# show ipv6 ospf database

Area ID         Type LS ID    Adv Rtr         Seq(Hex) Age  Cksum  Len
0               Link 000001e6 223.223.223.223 800000ab 1547 8955   68
0               Link 000000d8 1.1.1.1         800000aa 1295 0639   68
0               Link 00000185 223.223.223.223 800000ab 1481 7e6b   56
0               Iap  00000077 223.223.223.223 800000aa 1404 966a   56
0               Rtr  00000124 223.223.223.223 800000b0 1397 912c   40
0               Net  00000016 223.223.223.223 800000aa 1388 1b09   32
0               Iap  000001d1 223.223.223.223 800000bd 1379 a072   72
0               Iap  000000c3 1.1.1.1         800000ae 1325 e021   52
0               Rtr  00000170 1.1.1.1         800000ad 1280 af8e   40
N/A             Extn 00000062 223.223.223.223 800000ae 1409 0ca7   32
N/A             Extn 0000021d 223.223.223.223 800000a8 1319 441e   32
```

Syntax: show ipv6 ospf database [advrtr <ipv4-address> | as-external | extensive | inter-prefix | inter-router | intra-prefix | link | link-id <number> | network | router [scope <area-id> | as | link]]

The **advrtr** <ipv4-address> parameter displays detailed information about the LSAs for a specified advertising router only.

The **as-external** keyword displays detailed information about the AS externals LSAs only.

The **extensive** keyword displays detailed information about all LSAs in the database.

The **inter-prefix** keyword displays detailed information about the inter-area prefix LSAs only.

The **inter-router** keyword displays detailed information about the inter-area router LSAs only.

The **intra-prefix** keyword displays detailed information about the intra-area prefix LSAs only.

The **link** keyword displays detailed information about the link LSAs only.

The **link-id** <number> parameter displays detailed information about the specified link LSAs only.

The **network** <number> displays detailed information about the network LSAs only.

The **router** <number> displays detailed information about the router LSAs only.

The **scope** <area-id> parameter displays detailed information about the LSAs for a specified area, AS, or link.

This display shows the following information:

Table 36.2:OSPF V3 database summary fields

This Field...	Displays...
Area ID	The OSPF area in which the Foundry device resides.
Type	Type of LSA. LSA types can be the following: • Rtr – Router LSAs (Type 1). • Net – Network LSAs (Type 2). • Inap – Inter-area prefix LSAs for ABRs (Type 3). • Inar – Inter-area router LSAs for ASBRs (Type 4). • Extn – AS external LSAs (Type 5). • Link – Link LSAs (Type 8). • Iap – Intra-area prefix LSAs (Type 9).
LS ID	The ID of the LSA, in hexadecimal, from which the device learned this route.
Adv Rtr	The device that advertised the route.
Seq(Hex)	The sequence number of the LSA. The OSPF neighbor that sent the LSA stamps it with a sequence number to enable the Foundry device and other OSPF routers to determine which LSA for a given route is the most recent.
Age	The age of the LSA, in seconds.
Chksum	A checksum for the LSA packet. The checksum is based on all the fields in the packet except the age field. The Foundry device uses the checksum to verify that the packet is not corrupted.
Len	The length, in bytes, of the LSA.

For example, to display detailed information about all LSAs in the database, enter the following command at any CLI level:

```
FastIron# show ipv6 ospf database extensive
Area ID         Type LS ID    Adv Rtr         Seq(Hex) Age  Cksum  Len
0               Link 00000031 1.1.1.1         80000001 35   6db9   56
     Router Priority: 1
     Options: V6E---R--
     LinkLocal Address: fe80::1
     Number of Prefix: 1
     Prefix Options:
     Prefix: 3002::/64
    ...
Area ID         Type LS ID    Adv Rtr         Seq(Hex) Age  Cksum  Len
0               Iap  00000159 223.223.223.223 800000ab 357  946b   56
     Number of Prefix: 2
     Referenced LS Type: Network
     Referenced LS ID: 00000159
     Referenced Advertising Router: 223.223.223.223
     Prefix Options:  Metric: 0
     Prefix: 2000:4::/64
     Prefix Options:  Metric: 0
     Prefix: 2002:c0a8:46a::/64
Area ID         Type LS ID    Adv Rtr         Seq(Hex) Age  Cksum  Len
0               Rtr  00000039 223.223.223.223 800000b1 355  8f2d   40
  Capability Bits: --E-
  Options: V6E---R--
  Type: Transit Metric: 1
  Interface ID: 00000058  Neighbor Interface ID: 00000058
  Neighbor Router ID: 223.223.223.223
Area ID         Type LS ID    Adv Rtr         Seq(Hex) Age  Cksum  Len
0               Net  000001f4 223.223.223.223 800000ab 346  190a   32
     Options: V6E---R--
     Attached Router: 223.223.223.223
     Attached Router: 1.1.1.1
...
Area ID         Type LS ID    Adv Rtr         Seq(Hex) Age  Cksum  Len
N/A             Extn 000001df 223.223.223.223 800000af 368  0aa8   32
     Bits: E
     Metric: 00000001
     Prefix Options:
     Referenced LSType: 0
     Prefix: 2002::/16

Area ID         Type LS ID    Adv Rtr         Seq(Hex) Age  Cksum  Len
1               Inap 0000011d 10.1.1.188      80000001 124  25de   36
     Metric: 2
     Prefix Options:
     Prefix: 2000:2:2::/64

Area ID         Type LS ID    Adv Rtr         Seq(Hex) Age  Cksum  Len
0               Inar 0000005b 10.1.1.198      80000001 990  dbad   32
     Options: V6E---R--
     Metric: 1
     Destination Router ID:10.1.1.188
```

NOTE: Portions of this display are truncated for brevity. The purpose of this display is to show all possible fields that might display rather than to show complete output.

The fields that display depend upon the LSA type as shown in the following:

Table 36.3:OSPF V3 detailed database information fields

This Field...	Displays...
Router LSA (Type 1) (Rtr) Fields	
Capability Bits	A bit that indicates the capability of the Foundry device. The bit can be set to one of the following: • B – The device is an area border router. • E – The device is an AS boundary router. • V – The device is a virtual link endpoint. • W – The device is a wildcard multicast receiver.
Options	A 24-bit field that enables IPv6 OSPF routers to support the optional capabilities. When set, the following bits indicate the following: V6 – The device should be included in IPv6 routing calculations. E – The device floods AS-external-LSAs as described in RFC 2740. MC – The device forwards multicast packets as described in RFC 1586. N – The device handles type 7 LSAs as described in RFC 1584. R – The originator is an active router. DC –The device handles demand circuits.
Type	The type of interface. Possible types can be the following: • Point-to-point – A point-to-point connection to another router. • Transit – A connection to a transit network. • Virtual link – A connection to a virtual link.
Metric	The cost of using this router interface for outbound traffic.
Interface ID	The ID assigned to the router interface.
Neighbor Interface ID	The interface ID that the neighboring router has been advertising in hello packets sent on the attached link.
Neighbor Router ID	The router ID (IPv4 address) of the neighboring router that advertised the route. (By default, the Foundry router ID is the IPv4 address configured on the lowest numbered loopback interface. If the Foundry device does not have a loopback interface, the default router ID is the lowest numbered IPv4 address configured on the device.)

Table 36.3:OSPF V3 detailed database information fields (Continued)

This Field...	Displays...
Network LSA (Type 2) (Net) Fields	
Options	A 24-bit field that enables IPv6 OSPF routers to support the optional capabilities. When set, the following bits indicate the following: V6 – The device should be included in IPv6 routing calculations. E – The device floods AS-external-LSAs as described in RFC 2740. MC – The device forwards multicast packets as described in RFC 1586. N – The device handles type 7 LSAs as described in RFC 1584. R – The originator is an active router. DC –The device handles demand circuits.
Attached Router	The address of the neighboring router that advertised the route.
Inter-Area Prefix LSA (Type 3) (Inap) Fields	
Metric	The cost of the route.
Prefix Options	An 8-bit field describing various capabilities associated with the prefix.
Prefix	The IPv6 prefix included in the LSA.
Inter-Area Router LSA (Type 4) (Inar) Fields	
Options	A 24-bit field that enables IPv6 OSPF routers to support the optional capabilities. When set, the following bits indicate the following: V6 – The device should be included in IPv6 routing calculations. E – The device floods AS-external-LSAs as described in RFC 2740. MC – The device forwards multicast packets as described in RFC 1586. N – The device handles type 7 LSAs as described in RFC 1584. R – The originator is an active router. DC –The device handles demand circuits.
Metric	The cost of the route.
Destination Router ID	The ID of the router described in the LSA.
AS External LSA (Type 5) (Extn) Fields	
Bits	The bit can be set to one of the following: • E – If bit E is set, a Type 2 external metric. If bit E is zero, a Type 1 external metric. • F – A forwarding address is included in the LSA. • T – An external route tag is included in the LSA.
Metric	The cost of this route, which depends on bit E.
Prefix Options	An 8-bit field describing various capabilities associated with the prefix.

Table 36.3:OSPF V3 detailed database information fields (Continued)

This Field...	Displays...
Referenced LS Type	If non-zero, an LSA with this LS type is associated with the LSA.
Prefix	The IPv6 prefix included in the LSA.
Link LSA (Type 8) (Link) Fields	
Router Priority	The router priority of the interface attaching the originating router to the link.
Options	The set of options bits that the router would like set in the network LSA that will be originated for the link.
Link Local Address	The originating router's link-local interface address on the link.
Number of Prefix	The number of IPv6 address prefixes contained in the LSA.
Prefix Options	An 8-bit field of capabilities that serve as input to various routing calculations: • NU – The prefix is excluded from IPv6 unicast calculations. • LA – The prefix is an IPv6 interface address of the advertising router. • MC – The prefix is included in IPv6 multicast routing calculations. • P – NSSA area prefixes are readvertised at the NSSA area border.
Prefix	The IPv6 prefix included in the LSA.
Intra-Area Prefix LSAs (Type 9) (Iap) Fields	
Number of Prefix	The number of prefixes included in the LSA.
Referenced LS Type, Referenced LS ID	Identifies the router-LSA or network-LSA with which the IPv6 address prefixes are associated.
Referenced Advertising Router	The address of the neighboring router that advertised the route.
Prefix Options	An 8-bit field describing various capabilities associated with the prefix.
Metric	The cost of using the advertised prefix.
Prefix	The IPv6 prefix included in the LSA.
Number of Prefix	The number of prefixes included in the LSA.

Displaying OSPF V3 Interface Information

You can display a summary of information for all OSPF V3 interfaces or detailed information about a specified OSPF V3 interface.

To display a summary of OSPF V3 interfaces, enter the following command at any CLI level:

```
FastIron# show ipv6 ospf interface

Interface   OSPF      Status State     Area
ethe 3/1              up
ethe 3/2    enabled   up     DR        0
ethe 3/4    disabled  down
loopback 2 enabled   up     Loopback  0
tunnel 1    disabled  down
tunnel 2    enabled   up     P2P       0
tunnel 6              up
```

Syntax: show ipv6 ospf interface [ethernet <port> | loopback <number> | tunnel <number> | ve <number>]

The **ethernet | loopback | tunnel | ve** parameter specifies the interface for which to display information. If you specify an Ethernet interface, also specify the port number associated with the interface. If you specify a loopback, tunnel, or VE interface, also specify the number associated with the interface.

This display shows the following information:

Table 36.4:Summary of OSPF V3 interface information

This Field...	Displays...
Interface	The interface type, and the port number or number of the interface.
OSPF	The state of OSPF V3 on the interface. Possible states include the following: • Enabled. • Disabled.
Status	The status of the link. Possible status include the following: • Up. • Down.

Table 36.4:Summary of OSPF V3 interface information (Continued)

This Field...	Displays...
State	The state of the interface. Possible states includes the following: • DR – The interface is functioning as the Designated Router for OSPF V3. • BDR – The interface is functioning as the Backup Designated Router for OSPF V3. • Loopback – The interface is functioning as a loopback interface. • P2P – The interface is functioning as a point-to-point interface. • Passive – The interface is up but it does not take part in forming an adjacency. • Waiting – The interface is trying to determine the identity of the BDR for the network. • None – The interface does not take part in the OSPF interface state machine. • Down – The interface is unusable. No protocol traffic can be sent or received on such a interface. • DR other – The interface is a broadcast or NBMA network on which another router is selected to be the DR.
Area	The OSPF area to which the interface belongs.

For example, to display detailed information about Ethernet interface 2, enter the following command at any level of the CLI:

```
FastIron# show ipv6 ospf interface ethernet 3/2
ethe 3/2 is up, type BROADCAST
  IPv6 Address:
      2002:c0a8:46a::1/64
      2000:4::106/64
  Instance ID 0, Router ID 223.223.223.223
  Area ID 0, Cost 1
  State DR, Transmit Delay 1 sec, Priority 1
  Timer intervals :
    Hello 10, Dead 40, Retransmit 5
  DR:223.223.223.223 BDR:1.1.1.1  Number of I/F scoped LSAs is 2
  DRElection:     5 times, DelayedLSAck:   523 times
  Neighbor Count = 1,   Adjacent Neighbor Count= 1
     Neighbor:
      1.1.1.1 (BDR)
    Statistics of interface ethe 3/2:
      Type       tx   rx tx-byte rx-byte
      Unknown     0    0       0       0
      Hello    3149 3138 1259284 1255352
      DbDesc      7    6     416     288
      LSReq       2    2      80     152
      LSUpdate 1508  530  109508   39036
      LSAck     526 1398   19036   54568
```

This display shows the following information:

Table 36.5:Detailed OSPF V3 interface information

This Field...	Displays...
Interface status	The status of the interface. Possible status includes the following: • Up. • Down.
Type	The type of OSPF V3 circuit running on the interface. Possible types include the following: • BROADCAST • POINT TO POINT UNKNOWN
IPv6 Address	The IPv6 address(es) assigned to the interface.
Instance ID	An identifier for an instance of OSPF V3.
Router ID	The IPv4 address of the Foundry device. By default, the Foundry router ID is the IPv4 address configured on the lowest numbered loopback interface. If the device does not have a loopback interface, the default router ID is the lowest numbered IPv4 address configured on the device.
Area ID	The IPv4 address or numerical value of the area in which the interface belongs.
Cost	The overhead required to send a packet through the interface.
State	The state of the interface. Possible states include the following: • DR – The interface is functioning as the Designated Router for OSPF V3. • BDR – The interface is functioning as the Backup Designated Router for OSPF V3. • Loopback – The interface is functioning as a loopback interface. • P2P – The interface is functioning as a point-to-point interface. • Passive – The interface is up but it does not take part in forming an adjacency. • Waiting – The interface is trying to determine the identity of the BDR for the network. • None – The interface does not take part in the OSPF interface state machine. • Down – The interface is unusable. No protocol traffic can be sent or received on such a interface. • DR other – The interface is a broadcast or NBMA network on which another router is selected to be the DR.
Transmit delay	The amount of time, in seconds, it takes to transmit Link State Updates packets on the interface.

Table 36.5:Detailed OSPF V3 interface information (Continued)

This Field...	Displays...
Priority	The priority used when selecting the DR and the BDR. If the priority is 0, the interface does not participate in the DR and BDR election.
Timer intervals	The interval, in seconds, of the hello-interval, dead-interval, and retransmit-interval timers.
DR	The router ID (IPv4 address) of the DR.
BDR	The router ID (IPv4 address) of the BDR.
Number of I/F scoped LSAs	The number of interface LSAs scoped for a specified area, AS, or link.
DR Election	The number of times the DR election occurred.
Delayed LSA Ack	The number of the times the interface sent a delayed LSA acknowledgement.
Neighbor Count	The number of neighbors to which the interface is connected.
Adjacent Neighbor Count	The number of neighbors with which the interface has formed an active adjacency.
Neighbor	The router ID (IPv4 address) of the neighbor. This field also identifies the neighbor as a DR or BDR, if appropriate.
Interface statistics	The following statistics are provided for the interface: • Unknown – The number of Unknown packets transmitted and received by the interface. Also, the total number of bytes associated with transmitted and received Unknown packets. • Hello – The number of Hello packets transmitted and received by the interface. Also, the total number of bytes associated with transmitted and received Hello packets. • DbDesc – The number of Database Description packets transmitted and received by the interface. Also, the total number of bytes associated with transmitted and received Database Description packets. • LSReq – The number of link-state requests transmitted and received by the interface. Also, the total number of bytes associated with transmitted and received link-state requests. • LSUpdate – The number of link-state updates transmitted and received by the interface. Also, the total number of bytes associated with transmitted and received link-state requests. • LSAck – The number of link-state acknowledgements transmitted and received by the interface. Also, the total number of bytes associated with transmitted and received link-state acknowledgements.

Displaying OSPF V3 Memory Usage

To display information about OSPF V3 memory usage, enter the following command at any level of the CLI:

```
FastIron# show ipv6 ospf memory
Total Static Memory Allocated : 5829 bytes
Total Dynamic Memory Allocated : 0 bytes
Memory Type                Size       Allocated  Max-alloc  Alloc-Fails
MTYPE_OSPF6_TOP            0          0          0          0
MTYPE_OSPF6_LSA_HDR        0          0          0          0
MTYPE_OSPF6_RMAP_COMPILED  0          0          0          0
MTYPE_OSPF6_OTHER          0          0          0          0
MTYPE_THREAD_MASTER        0          0          0          0
MTYPE_OSPF6_AREA           0          0          0          0
MTYPE_OSPF6_AREA_RANGE     0          0          0          0
MTYPE_OSPF6_SUMMARY_ADDRE  0          0          0          0
MTYPE_OSPF6_IF             0          0          0          0
MTYPE_OSPF6_NEIGHBOR       0          0          0          0
MTYPE_OSPF6_ROUTE_NODE     0          0          0          0
MTYPE_OSPF6_ROUTE_INFO     0          0          0          0
MTYPE_OSPF6_PREFIX         0          0          0          0
MTYPE_OSPF6_LSA            0          0          0          0
MTYPE_OSPF6_VERTEX         0          0          0          0
MTYPE_OSPF6_SPFTREE        0          0          0          0
MTYPE_OSPF6_NEXTHOP        0          0          0          0
MTYPE_OSPF6_EXTERNAL_INFO  0          0          0          0
MTYPE_THREAD               0          0          0          0
```

Syntax: show ipv6 ospf memory

This display shows the following information:

Table 36.6:OSPF V3 memory usage information

This Field...	Displays...
Total Static Memory Allocated	A summary of the amount of static memory allocated, in bytes, to OSPF V3.
Total Dynamic Memory Allocated	A summary of the amount of dynamic memory allocated, in bytes, to OSPF V3.
Memory Type	The type of memory used by OSPF V3. (This information is for use by Foundry's technical support in case of a problem.)
Size	The size of a memory type.
Allocated	The amount of memory currently allocated to a memory type.
Max-alloc	The maximum amount of memory that was allocated to a memory type.
Alloc-Fails	The number of times an attempt to allocate memory to a memory type failed.

Displaying OSPF V3 Neighbor Information

You can display a summary of OSPF V3 neighbor information for the Foundry device or detailed information about a specified neighbor.

To display a summary of OSPF V3 neighbor information for the device, enter the following command at any CLI level:

```
FastIron# show ipv6 ospf neighbor
RouterID        Pri State    DR              BDR             Interface[State]
1.1.1.1           1 Full     223.223.223.223 1.1.1.1         ethe 3/2   [DR]
```

Syntax: show ipv6 ospf neighbor [router-id <ipv4-address>]

The **router-id** <ipv4-address> parameter displays only the neighbor entries for the specified router.

This display shows the following information:

Table 36.7:Summary of OSPF V3 neighbor information

Field	Description
Router ID	The IPv4 address of the neighbor. By default, the Foundry router ID is the IPv4 address configured on the lowest numbered loopback interface. If the device does not have a loopback interface, the default router ID is the lowest numbered IPv4 address configured on the device.
Pri	The OSPF V3 priority of the neighbor. The priority is used during election of the DR and BDR.
State	The state between the Foundry device and the neighbor. The state can be one of the following: • Down • Attempt • Init • 2-Way • ExStart • Exchange • Loading • Full
DR	The router ID (IPv4 address) of the DR.
BDR	The router ID (IPv4 address) of the BDR.

Table 36.7:Summary of OSPF V3 neighbor information (Continued)

Field	Description
Interface [State]	The interface through which the router is connected to the neighbor. The state of the interface can be one of the following: • DR – The interface is functioning as the Designated Router for OSPF V3. • BDR – The interface is functioning as the Backup Designated Router for OSPF V3. • Loopback – The interface is functioning as a loopback interface. • P2P – The interface is functioning as a point-to-point interface. • Passive – The interface is up but it does not take part in forming an adjacency. • Waiting – The interface is trying to determine the identity of the BDR for the network. • None – The interface does not take part in the OSPF interface state machine. • Down – The interface is unusable. No protocol traffic can be sent or received on such a interface. • DR other – The interface is a broadcast or NBMA network on which another router is selected to be the DR.

For example, to display detailed information about a neighbor with the router ID of 1.1.1.1, enter the following command at any CLI level:

```
FastIron# show ipv6 ospf neighbor router-id 3.3.3.3
RouterID        Pri State    DR              BDR             Interface[State]
3.3.3.3           1 Full     3.3.3.3         1.1.1.1         ve 10    [BDR]
DbDesc bit for this neighbor: --s
Nbr Ifindex of this router: 1
Nbr DRDecision: DR 3.3.3.3, BDR 1.1.1.1
Last received DbDesc: opt:xxx ifmtu:0 bit:--s seqnum:0
Number of LSAs in DbDesc retransmitting: 0
Number of LSAs in SummaryList: 0
Number of LSAs in RequestList: 0
Number of LSAs in RetransList: 0
SeqnumMismatch 0 times, BadLSReq 0 times
OnewayReceived 0 times, InactivityTimer 0 times
DbDescRetrans 0 times, LSReqRetrans 0 times
LSUpdateRetrans 1 times
LSAReceived 12 times, LSUpdateReceived 6 times
```

This display shows the following information:

Table 36.8:Detailed OSPF V3 neighbor information

Field	Description
Router ID	For information about this field, see Table 36.7 on page 36-29.
Pri	For information about this field, see Table 36.7 on page 36-29.
State	For information about this field, see Table 36.7 on page 36-29.

Table 36.8:Detailed OSPF V3 neighbor information (Continued)

Field	Description
DR	For information about this field, see Table 36.7 on page 36-29.
BDR	For information about this field, see Table 36.7 on page 36-29.
Interface [State]	For information about this field, see Table 36.7 on page 36-29.
DbDesc bit...	The Database Description packet, which includes 3 bits of information: • The first bit can be “i” or “-”. “i” indicates the inet bit is set. “-” indicates the inet bit is not set. • The second bit can be “m” or “-”. “m” indicates the more bit is set. “-” indicates the more bit is not set. • The third bit can be “m” or “s”. An “m” indicates the master. An “s” indicates standby.
Index	The ID of the LSA from which the neighbor learned of the router.
DR Decision	The router ID (IPv4 address) of the neighbor’s elected DR and BDR.
Last Received Db Desc	The content of the last database description received from the specified neighbor.
Number of LSAs in Db Desc retransmitting	The number of LSAs that need to be retransmitted to the specified neighbor.
Number of LSAs in Summary List	The number of LSAs in the neighbor’s summary list.
Number of LSAs in Request List	The number of LSAs in the neighbor’s request list.
Number of LSAs in Retransmit List	The number of LSAs in the neighbor’s retransmit list.
Seqnum Mismatch	The number of times sequence number mismatches occurred.
BadLSReq	The number of times the neighbor received a bad link-state request from the Foundry device.
One way received	The number of times a hello packet, which does not mention the router, is received from the neighbor. This omission in the hello packet indicates that the communication with the neighbor is not bidirectional.
Inactivity Timer	The number of times that the neighbor’s inactivity timer expired.
Db Desc Retransmission	The number of times sequence number mismatches occurred.
LSReqRetrans	The number of times the neighbor retransmitted link-state requests to the Foundry device.
LSUpdateRetrans	The number of times the neighbor retransmitted link-state updates to the Foundry device.
LSA Received	The number of times the neighbor received LSAs from the Foundry device.
LS Update Received	The number of times the neighbor received link-state updates from the Foundry device.

Displaying Routes Redistributed into OSPF V3

You can display all IPv6 routes or a specified IPv6 route that the Foundry device has redistributed into OSPF V3.

To display all IPv6 routes that the device has redistributed into OSPF V3, enter the following command at any level of the CLI:

```
FastIron# show ipv6 ospf redistribute route
Id      Prefix                                        Protocol  Metric Type  Metric
 snIpAsPathAccessListStringRegExpression
1       2002::/16                                     Static    Type-2       1
2       2002:1234::/32                                Static    Type-2       1
```

Syntax: show ipv6 ospf redistribute route [<ipv6-prefix>]

The <ipv6-prefix> parameter specifies an IPv6 network prefix. (You do not need to specify the length of the prefix.)

For example, to display redistribution information for the prefix 2002::, enter the following command at any level of the CLI:

```
FastIron# show ipv6 ospf redistribute route 2002::
Id      Prefix                                        Protocol  Metric Type  Metric
1       2002::/16                                     Static    Type-2       1
```

These displays show the following information:

Table 36.9:OSPF V3 redistribution information

This Field...	Displays...
ID	An ID for the redistributed route.
Prefix	The IPv6 routes redistributed into OSPF V3.
Protocol	The protocol from which the route is redistributed into OSPF V3. Redistributed protocols can be the following: • RIP – RIPng. • Static – IPv6 static route table. • Connected – A directly connected network.
Metric Type	The metric type used for routes redistributed into OSPF V3. The metric type can be the following: • Type-1 – Specifies a small metric (2 bytes). • Type-2 – Specifies a big metric (3 bytes).
Metric	The value of the default redistribution metric, which is the OSPF cost of redistributing the route into OSPF V3.

Displaying OSPF V3 Route Information

You can display the entire OSPF V3 route table for the Foundry device or only the route entries for a specified destination.

To display the entire OSPF V3 route table for the device, enter the following command at any level of the CLI:

```
  FastIron# show ipv6 ospf routes
Current Route count: 4
    Intra: 4 Inter: 0 External: 0 (Type1 0/Type2 0)
    Equal-cost multi-path: 0
    Destination                    Options   Area             Cost Type2 Cost
    Next Hop Router                Outgoing Interface
*IA 2000:4::/64                    V6E---R-- 0.0.0.0             1 0
    ::                             ethe 3/2
*IA 2002:c0a8:46a::/64             V6E---R-- 0.0.0.0             1 0
    ::                             ethe 3/2
*IA 2999::1/128                    --------- 0.0.0.0             0 0
    ::                             loopback 2
*IA 2999::2/128                    V6E---R-- 0.0.0.0             1 0
    fe80::2e0:52ff:fe91:bb37       ethe 3/2
```

Syntax: show ipv6 ospf routes [<ipv6-prefix>]

The <ipv6-prefix> parameter specifies a destination IPv6 prefix. (You do not need to specify the length of the prefix.) If you use this parameter, only the route entries for this destination are shown.

For example, to display route information for the destination prefix 2000:4::, enter the following command at any level of the CLI:

```
 FastIron# show ipv6 ospf routes 2000:4::
 Destination                    Options   Area             Cost Type2 Cost
    Next Hop Router                Outgoing Interface
*IA 2000:4::/64                    V6E---R-- 0.0.0.0             1 0
    ::                             ethe 3/2
```

These displays show the following information:

Table 36.10:OSPF V3 route information

This Field...	Displays...
Current Route Count (Displays with the entire OSPF V3 route table only)	The number of route entries currently in the OSPF V3 route table.
Intra/Inter/External (Type1/Type2) (Displays with the entire OSPF V3 route table only)	The breakdown of the current route entries into the following route types: • Inter – The number of routes that pass into another area. • Intra – The number of routes that are within the local area. • External1 – The number of type 1 external routes. • External2 – The number of type 2 external routes.
Equal-cost multi-path (Displays with the entire OSPF V3 route table only)	The number of equal-cost routes to the same destination in the OSPF V3 route table. If load sharing is enabled, the router equally distributes traffic among the routes.
Destination	The IPv6 prefixes of destination networks to which the Foundry device can forward IPv6 packets. “*IA” indicates the next router is an intra-area router.

Table 36.10:OSPF V3 route information (Continued)

This Field...	Displays...
Options	A 24-bit field that enables IPv6 OSPF routers to support the optional capabilities. When set, the following bits indicate the following: V6 – The device should be included in IPv6 routing calculations. E – The device floods AS-external-LSAs as described in RFC 2740. MC – The device forwards multicast packets as described in RFC 1586. N – The device handles type 7 LSAs as described in RFC 1584. R – The originator is an active router. DC –The device handles demand circuits.
Area	The area whose link state information has led to the routing table entry's collection of paths.
Cost	The type 1 cost of this route.
Type2 Cost	The type 2 cost of this route.
Next-Hop Router	The IPv6 address of the next router a packet must traverse to reach a destination.
Outgoing Interface	The router interface through which a packet must traverse to reach the next-hop router.

Displaying OSPF V3 SPF Information

You can display the following OSPF V3 SPF information:

- SPF node information for a specified area.
- SPF table for a specified area.
- SPF tree for a specified area.

For example, to display information about SPF nodes in area 0, enter the following command at any level of the CLI:

```
FastIron# show ipv6 ospf spf node area 0
SPF node for Area 0
SPF node 223.223.223.223,  cost: 0,  hops: 0
 nexthops to node:
 parent nodes:
 child nodes: 223.223.223.223:88

SPF node 223.223.223.223:88,  cost: 1,  hops: 1
 nexthops to node:    :: ethe 3/2
 parent nodes: 223.223.223.223
 child nodes: 1.1.1.1:0

SPF node 1.1.1.1:0,  cost: 1,  hops: 2
 nexthops to node:    fe80::2e0:52ff:fe91:bb37 ethe 3/2
 parent nodes: 223.223.223.223:88
 child nodes:
```

Syntax: show ipv6 ospf spf node area [<area-id>]

The **node** keyword displays SPF node information.

The **area** <area-id> parameter specifies a particular area. You can specify the <area-id> in the following formats:

- As an IPv4 address; for example, 192.168.1.1.
- As a numerical value from 0 – 2,147,483,647.

This display shows the following information:

Table 36.11:OSPF V3 SPF node information

This Field...	Displays...
SPF node	Each SPF node is identified by its router ID (IPv4 address). If the node is a child node, it is additionally identified by an interface on which the node can be reached appended to the router ID in the format <router-id>:<interface-id>.
Cost	The cost of traversing the SPF node to reach the destination.
Hops	The number of hops needed to reach the parent SPF node.
Next Hops to Node	The IPv6 address of the next hop-router and/or the router interface through which to access the next-hop router.
Parent Nodes	The SPF node's parent nodes. A ***parent node*** is an SPF node at the highest level of the SPF tree, which is identified by its router ID.
Child Nodes	The SPF node's child nodes. A ***child node*** is an SPF node at a lower level of the SPF tree, which is identified by its router ID and interface on which the node can be reached.

For example, to display the SPF table for area 0, enter the following command at any level of the CLI:

```
FastIron# show ipv6 ospf spf table area 0
  SPF table for Area 0
  Destination         Bits Options  Cost  Nexthop                  Interface
R 1.1.1.1             ---- V6E---R-    1  fe80::2e0:52ff:fe91:bb37  ethe 3/2
N 223.223.223.223[88] ---- V6E---R-    1  ::                        ethe 3/2
```

Syntax: show ipv6 ospf spf table area <area-id>

The **table** parameter displays the SPF table.

The **area** <area-id> parameter specifies a particular area. You can specify the <area-id> in the following formats:

- As an IPv4 address, for example, 192.168.1.1.
- As a numerical value from 0 – 2,147,483,647.

This display shows the following information:

Table 36.12:OSPF V3 SPF Table

This Field...	Displays...
Destination	The destination of a route, which is identified by the following: • "R", which indicates the destination is a router. "N", which indicates the destination is a network. • An SPF node's router ID (IPv4 address). If the node is a child node, it is additionally identified by an interface on which the node can be reached appended to the router ID in the format <router-id>:<interface-id>.
Bits	A bit that indicates the capability of the Foundry device. The bit can be set to one of the following: • B – The device is an area border router. • E – The device is an AS boundary router. • V – The device is a virtual link endpoint. • W – The device is a wildcard multicast receiver.
Options	A 24-bit field that enables IPv6 OSPF routers to support the optional capabilities. When set, the following bits indicate the following: V6 – The router should be included in IPv6 routing calculations. E – The router floods AS-external-LSAs as described in RFC 2740. MC – The router forwards multicast packets as described in RFC 1586. N – The router handles type 7 LSAs as described in RFC 1584. R – The originator is an active router. DC –The router handles demand circuits.
Cost	The cost of traversing the SPF node to reach the destination.
Next hop	The IPv6 address of the next hop-router.
Interface	The router interface through which to access the next-hop router.

For example, to display the SPF tree for area 0, enter the following command at any level of the CLI:

```
FastIron# show ipv6 ospf spf tree area 0
   SPF tree for Area 0
  +- 223.223.223.223 cost 0
      +- 223.223.223.223:88 cost 1
          +- 1.1.1.1:0 cost 1
```

Syntax: show ipv6 ospf spf tree area <area-id>

The **tree** keyword displays the SPF table.

The **area** <area-id> parameter specifies a particular area. You can specify the <area-id> in the following formats:

- As an IPv4 address; for example, 192.168.1.1.
- As a numerical value from 0 – 2,147,483,647.

In this sample output, consider the SPF node with the router ID 223.223.223.223 to be the top (root) of the tree and the local router. Consider all other layers of the tree (223.223.223.223:88 and 1.1.1.1:0) to be destinations in the network. Therefore, traffic destined from router 223.223.223.223 to router 1.1.1.1:0 must first traverse router 223.223.223.223:88.

Displaying IPv6 OSPF Virtual Link Information

To display OSPF V3 virtual link information for the Foundry device, enter the following command at any level of the CLI:

```
FastIron# show ipv6 ospf virtual-link
Index Transit Area ID  Router ID        Interface Address            State
1     1                1.1.1.1          3003::2                      P2P
```

Syntax: show ipv6 ospf virtual-link

This display shows the following information:

Table 36.13:OSPF V3 virtual link information

This Field...	Displays...
Index	An index number associated with the virtual link.
Transit Area ID	The ID of the shared area of two ABRs that serves as a connection point between the two routers.
Router ID	IPv4 address of the router at the other end of the virtual link (virtual neighbor).
Interface Address	The local address used to communicate with the virtual neighbor.
State	The state of the virtual link. Possible states include the following: • P2P – The link is functioning as a point-to-point interface. • DOWN – The link is down.

Displaying OSPF V3 Virtual Neighbor Information

To display OSPF V3 virtual neighbor information for the Foundry device, enter the following command at any level of the CLI:

```
FastIron# show ipv6 ospf virtual-neighbor
Index Router ID        Address                      State     Interface
1     1.1.1.1          3002::1                      Full      ethe 2/3
```

Syntax: show ipv6 ospf virtual-neighbor

This display shows the following information:

Table 36.14:OSPF V3 virtual neighbor information

This Field...	Displays...
Index	An index number associated with the virtual neighbor.
Router ID	IPv4 address of the virtual neighbor.
Address	The IPv6 address to be used for communication with the virtual neighbor.
State	The state between the Foundry device and the virtual neighbor. The state can be one of the following: • Down • Attempt • Init • 2-Way • ExStart • Exchange • Loading • Full
Interface	The IPv6 address of the virtual neighbor.

Chapter 37

Configuring VRRP and VRRPE

This chapter describes how to configure Foundry Layer 3 Switches with the following router redundancy protocols:

- ***Virtual Router Redundancy Protocol (VRRP)*** – The standard router redundancy protocol described in RFC 2338.
- ***VRRP Extended (VRRPE)*** – An enhanced version of VRRP that overcomes limitations in the standard protocol.

NOTE: VRRP and VRRPE are separate protocols. You cannot use them together.

NOTE: You can use a Foundry Layer 3 Switch configured for VRRP with another Foundry Layer 3 Switch or a third-party router that is also configured for VRRP. However, you can use a Foundry Layer 3 Switch configured for VRRPE only with another Foundry Layer 3 Switch that also is configured for VRRPE.

For a summary of how these two router redundancy protocols differ, see “Comparison of VRRP and VRRPE” on page 37-7.

Overview

The following sections describe VRRP and VRRPE. The protocols both provide redundant paths for IP addresses. However, the protocols differ in a few important ways. For clarity, each protocol is described separately.

Overview of VRRP

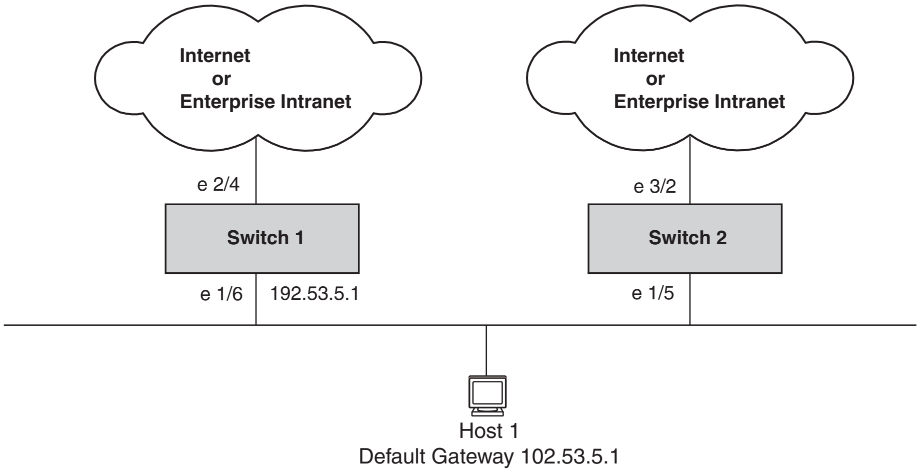
Platform Support:

- FESX and FSX running pre-release 02.4.00 software – L3
- FESX and FSX running software release 02.4.00 and later – BL3, L3
- FGS and FLS devices running software release 04.0.00 and later – BL3

NOTE: VRRP support in the base Layer 3 code is the same as in the full Layer 3 code.

VRRP is a protocol that provides redundancy to routers within a LAN. VRRP allows you to provide alternate router paths for a host without changing the IP address or MAC address by which the host knows its gateway. Consider the situation shown in Figure 37.1.

Figure 37.1 Switch 1 is Host1's Default Gateway but is a Single Point of Failure

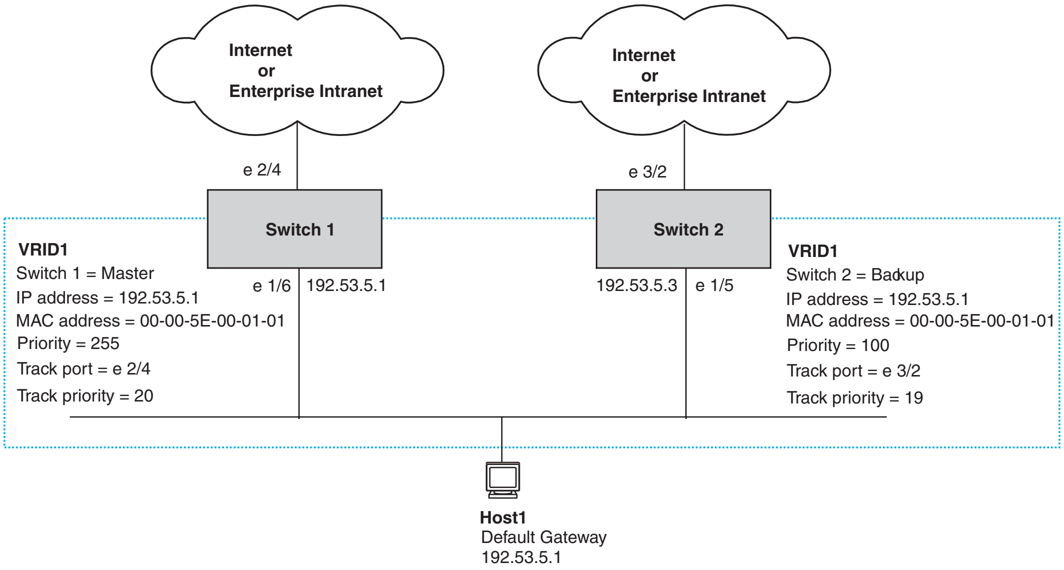


Switch 1 as the host's default gateway out of the subnet. If this interface goes down, Host1 is cut off from the rest of the network. Switch 1 is thus a single point of failure for Host1's access to other networks.

If Switch 1 fails, you could configure Host1 to use Switch 2. Configuring one host with a different default gateway might not require too much extra administration. However, consider a more realistic network with dozens or even hundreds of hosts per subnet; reconfiguring the default gateways for all the hosts is impractical. It is much simpler to configure a VRRP virtual router on Switch 1 and Switch 2 to provide a redundant path for the hosts.

Figure 37.2 shows the same example network shown in Figure 37.1, but with a VRRP virtual router configured on Switch 1 and Switch 2.

Figure 37.2 Switch 1 and Switch 2 are Configured as a VRRP Virtual Router for Redundant Network Access for Host1



The dashed box in Figure 37.2 represents a VRRP virtual router. When you configure a virtual router, one of the configuration parameters is the virtual router ID (VRID), which can be a number from 1 – 255. In this example, the VRID is 1.

NOTE: You can provide more redundancy by also configuring a second VRID with Switch 2 as the Owner and Switch 1 as the Backup. This type of configuration is sometimes called Multigroup VRRP.

Virtual Router ID (VRID)

A ***VRID*** consists of one Master router and one or more Backup routers. The Master router is the router that owns the IP address(es) you associate with the VRID. For this reason, the Master router is sometimes called the "Owner". Configure the VRID on the router that owns the default gateway interface. The other router in the VRID does not own the IP address(es) associated with VRID but provides the backup path if the Master router becomes unavailable.

Virtual Router MAC Address

Notice the MAC address associated with VRID1. The first five octets of the address are the standard MAC prefix for VRRP packets, as described in RFC 2338. The last octet is the VRID. THE VRID number becomes the final octet in the virtual MAC address associated with the virtual router.

When you configure a VRID, the software automatically assigns its MAC address. When a VRID becomes active, the Master router broadcasts a gratuitous ARP request containing the virtual router's MAC address for each IP address associated with the virtual router. In Figure 37.2, Switch 1 sends a gratuitous ARP with MAC address 00-00-5e-00-01-01 and IP address 192.53.5.1. Hosts use the virtual router's MAC address in routed traffic they send to their default IP gateway (in this example, 192.53.5.1).

Virtual Router IP Address

VRRP does not use virtual IP addresses. Thus, there is no virtual IP address associated with a virtual router. Instead, you associate the virtual router with one or more real interface IP addresses configured on the router that owns the real IP address(es). In Figure 37.2, the virtual router with VRID1 is associated with real IP address 192.53.5.1, which is configured on interface e1/6 on Switch 1. VRIDs are interface-level parameters, not system-level parameters, so the IP address you associate with the VRID must already be a real IP address configured on the Owner's interface.

NOTE: You also can associate a virtual router with a virtual interface. A virtual interface is a named set of physical interfaces.

When you configure the Backup router for the VRID, specify the same IP address as the one you specify on the Owner. This is the IP address used by the host as its default gateway. The IP address cannot also exist on the Backup router. The interface on which you configure the VRID on the Backup router must have an IP address in the same subnet.

NOTE: If you delete a real IP address used by a VRRP entry, the VRRP entry also is deleted automatically.

NOTE: When a Backup router takes over forwarding responsibilities from a failed Master router, the Backup forwards traffic addressed to the VRID MAC address, which the host believes is the MAC address of the router interface for its default gateway. However, the Backup cannot reply to IP pings sent to the IP address(es) associated with the VRID. Because the IP address(es) are owned by the Owner, if the Owner is unavailable, the IP addresses are unavailable as packet destinations.

Master Negotiation

The routers within a VRID use the VRRP priority values associated with each router to determine which router becomes the Master. When you configure the VRID on a router interface, you specify whether the router is the Owner of the IP addresses you plan to associate with the VRID or a Backup. If you indicate that the router is the Owner of the IP addresses, the software automatically sets the router's VRRP priority for the VRID to 255, the highest VRRP priority. The router with the highest priority becomes the Master.

Backup routers can have a priority from 3 – 254, which you assign when you configure the VRID on the Backup router's interfaces. The default VRRP priority for Backup routers is 100.

Because the router that owns the IP addresses associated with the VRID always has the highest priority, when all the routers in the virtual router are operating normally, the negotiation process results in the Owner of the VRID's IP addresses becoming the Master router. Thus, the VRRP negotiation results in the normal case, in which the hosts' path to the default route is to the router that owns the interface for that route.

Hello Messages

VRRP routers use Hello messages for negotiation to determine the Master router. VRRP routers send Hello messages to IP Multicast address 224.0.0.18. The frequency with which the Master sends Hello messages is the Hello Interval. Only the Master sends Hello messages. However, a Backup uses the Hello interval you configure for the Backup if it becomes the Master.

The Backup routers wait for a period of time called the Dead Interval for a Hello message from the Master. If a Backup router does not receive a Hello message by the time the dead interval expires, the Backup router assumes that the Master router is dead and negotiates with the other Backups to select a new Master router. The Backup router with the highest priority becomes the new Master.

If the Owner becomes unavailable, but then comes back online, the Owner again becomes the Master router. The Owner becomes the Master router again because it has the highest priority. The Owner always becomes the Master again when the Owner comes back online.

NOTE: If you configure a track port on the Owner and the track port is down, the Owner's priority is changed to the track priority. In this case, the Owner does not have a higher priority than the Backup that is acting as Master and the Owner therefore does not resume its position as Master. For more information about track ports, see "Track Ports and Track Priority" on page 37-4.

By default, if a Backup is acting as the Master, and the Master is still unavailable, another Backup can "preempt" the Backup that is acting as the Master. This can occur if the new Backup has a higher priority than the Backup who is acting as Master. You can disable this behavior if you want. When you disable preemption, a Backup router that has a higher priority than the router who is currently acting as Master does not preempt the new Master by initiating a new Master negotiation. See "Backup Preempt" on page 37-17.

NOTE: Regardless of the setting for the preempt parameter, the Owner always becomes the Master again when it comes back online.

Track Ports and Track Priority

The Foundry implementation of VRRP enhances the protocol by giving a VRRP router the capability to monitor the state of the interfaces on the other end of the route path through the router. For example, in Figure 37.2 on page 37-2, interface e1/6 on Switch 1 owns the IP address to which Host1 directs route traffic on its default gateway. The exit path for this traffic is through Router1's e2/4 interface.

Suppose interface e2/4 goes down. Even if interface e1/6 is still up, Host1 is nonetheless cut off from other networks. In conventional VRRP, Switch 1 would continue to be the Master router despite the unavailability of the exit interface for the path the router is supporting. However, if you configure interface e1/6 to track the state of interface e2/4, if e2/4 goes down, interface e1/6 responds by changing Switch 1's VRRP priority to the value of the track priority. In the configuration shown in Figure 37.2 on page 37-2, Switch 1's priority changes from 255 to 20. One of the parameters contained in the Hello messages the Master router sends to its Backups is the Master router's priority. If the track port feature results in a change in the Master router's priority, the Backup routers quickly become aware of the change and initiate a negotiation for Master router.

In Figure 37.2 on page 37-2, the track priority results in Switch 1's VRRP priority becoming lower than Switch 2's VRRP priority. As a result, when Switch 2 learns that it now has a higher priority than Switch 1, Switch 2 initiates negotiation for Master router and becomes the new Master router, thus providing an open path for Host1's traffic. To take advantage of the track port feature, make sure the track priorities are always lower than the VRRP priorities. The default track priority for the router that owns the VRID IP addresses is 2. The default track priority for Backup routers is 1. If you change the track port priorities, make sure you assign a higher track priority to the Owner of the IP addresses than the track priority you assign on the Backup routers.

Suppression of RIP Advertisements for Backed Up Interfaces

The Foundry implementation also enhances VRRP by allowing you to configure the protocol to suppress RIP advertisements for the backed up paths from Backup routers. Normally, a VRRP Backup router includes route information for the interface it is backing up in RIP advertisements. As a result, other routers receive multiple paths for the interface and might sometimes unsuccessfully use the path to the Backup rather than the path to the Master. If you enable the Foundry implementation of VRRP to suppress the VRRP Backup routers from advertising the backed up interface in RIP, other routers learn only the path to the Master router for the backed up interface.

Authentication

The Foundry implementation of VRRP can use simple passwords to authenticate VRRP packets. The VRRP authentication type is not a parameter specific to the VRID. Instead, VRRP uses the authentication type associated with the interfaces on which you define the VRID. For example, if you configure your router interfaces to use a simple password to authenticate traffic, VRRP uses the same simple password and VRRP packets that do not contain the password are dropped. If your interfaces do not use authentication, neither does VRRP.

NOTE: The MD5 authentication type is not supported for VRRP.

Independent Operation of VRRP alongside RIP, OSPF, and BGP4

VRRP operation is independent of the RIP, OSPF, and BGP4 protocols. Their operation is unaffected when VRRP is enabled on a RIP, OSPF, or BGP4 interface.

Dynamic VRRP Configuration

All VRRP global and interface parameters take effect immediately. You do not need to reset the system to place VRRP configuration parameters into effect.

Overview of VRRPE

Platform Support:

- FESX and FSX devices running software release 02.0.00 and later – L3

  NOTE: VRRPE is not supported in the base Layer 3 code.

VRRPE is similar to VRRP, but differs in the following respects:

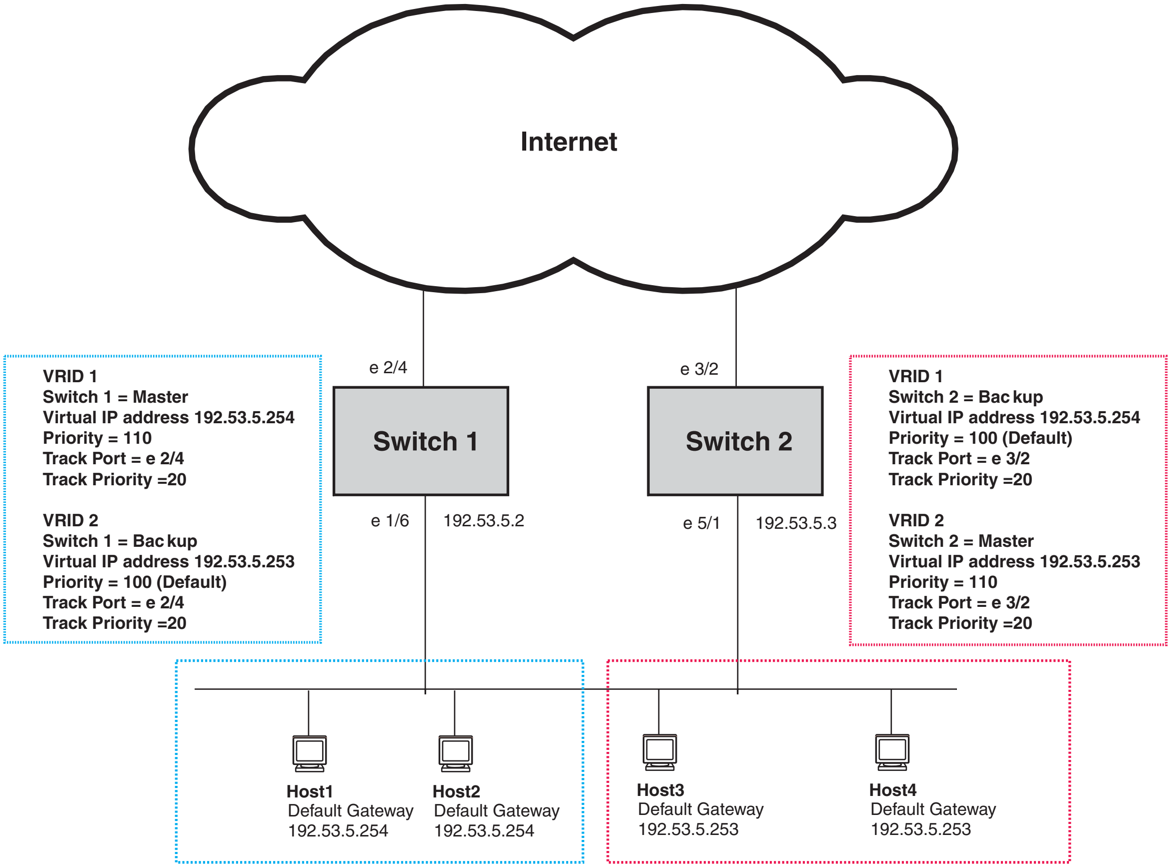
- Owners and Backups
  - VRRP has an Owner and one or more Backups for each VRID. The Owner is the router on which the VRID's IP address is also configured as a real address. All the other routers supporting the VRID are Backups.
  - VRRPE does not use Owners. All routers are Backups for a given VRID. The router with the highest priority becomes Master. If there is a tie for highest priority, the router with the highest IP address becomes Master. The elected Master owns the virtual IP address and answers ping and ARP requests and so on.
- VRID's IP address
  - VRRP requires that the VRID also be a real IP address configured on the VRID's interface on the Owner.
  - VRRPE requires only that the VRID be in the same subnet as an interface configured on the VRID's interface. In fact, VRRPE does not allow you to specify a real IP address configured on the interface as the VRID IP address.
- VRID's MAC Address

  - VRRP source MAC is a virtual MAC address defined as 00-00-5E-00-01-<vrid>, where <vrid> is the VRID. The Master owns the Virtual MAC address.
  - VRRPE uses the interface's actual MAC address as the source MAC address. The MAC address is 02-E0-52-<hash-value>-<vrid>, where <hash-value> is a two-octet hashed value for the IP address and <vrid> is the VRID.
- Hello packets
  - VRRP sends Hello messages to IP Multicast address 224.0.0.18.
  - VRRPE uses UDP to send Hello messages in IP multicast messages. The Hello packets use the interface's actual MAC address and IP address as the source addresses. The destination MAC address is 01-00-5E-00-00-02, and the destination IP address is 224.0.0.2 (the well-known IP multicast address for "all routers"). Both the source and destination UDP port number is 8888. VRRP messages are encapsulated in the data portion of the packet.
- Track ports and track priority
  - VRRP changes the priority of the VRID to the track priority, which typically is lower than the VRID priority and lower than the VRID's priorities configured on the Backups. For example, if the VRRP interface's priority is 100 and a tracked interface with track priority 20 goes down, the software changes the VRRP interface's priority to 20.
  - VRRPE reduces the priority of a VRRPE interface by the amount of a tracked interface's priority if the tracked interface's link goes down. For example, if the VRRPE interface's priority is 200 and a tracked interface with track priority 20 goes down, the software changes the VRRPE interface's priority to 180. If another tracked interface goes down, the software reduces the VRID's priority again, by the amount of the tracked interface's track priority.

The most important difference is that all VRRPE routers are Backups. There is no Owner router. VRRPE overcomes the limitations in standard VRRP by removing the Owner.

Figure 37.3 shows an example of a VRRPE configuration.

Figure 37.3 Router1 and Router2 are Configured to Provide Dual Redundant Network Access for the Host



In this example, Switch 1 and Switch 2 use VRRPE to load share as well as provide redundancy to the hosts. The load sharing is accomplished by creating two VRRPE groups. Each group has its own virtual IP addresses. Half of the clients point to VRID 1's virtual IP address as their default gateway and the other half point to VRID 2's virtual IP address as their default gateway. This will enable some of the outbound Internet traffic to go through Switch 1 and the rest to go through Switch 2.

Switch 1 is the master for VRID 1 (backup priority = 110) and Switch 2 is the backup for VRID 1 (backup priority = 100). Switch 1 and Switch 2 both track the uplinks to the Internet. If an uplink failure occurs on Switch 1, its backup priority is decremented by 20 (track priority = 20), so that all traffic destined to the Internet is sent through Switch 2 instead.

Similarly, Switch 2 is the master for VRID 2 (backup priority = 110) and Switch 1 is the backup for VRID 2 (backup priority = 100). Switch 1 and Switch 2 are both tracking the uplinks to the Internet. If an uplink failure occurs on Switch 1, its backup priority is decremented by 20 (track priority = 20), so that all traffic destined to the internet is sent through Switch 2 instead.

Configuration Note

VRRP-E is supported in the full Layer 3 code only. It is not supported in the Base Layer 3 code.

Comparison of VRRP and VRRPE

This section compares Foundry’s router redundancy protocols.

VRRP

VRRP is a standards-based protocol, described in RFC 2338. The Foundry implementation of VRRP contains the features in RFC 2338. The Foundry implementation also provides the following additional features:

- Track ports – A Foundry feature that enables you to diagnose the health of all the Layer 3 Switch's ports used by the backed-up VRID, instead of only the port connected to the client subnet. See "Track Ports and Track Priority" on page 37-4.
- Suppression of RIP advertisements on Backup routes for the backed up interface – You can enable the Layer 3 Switches to advertise only the path to the Master router for the backed up interface. Normally, a VRRP Backup router includes route information for the interface it is backing up in RIP advertisements.

Foundry Layer 3 Switches configured for VRRP can interoperate with third-party routers using VRRP.

VRRPE

VRRPE is a Foundry protocol that provides the benefits of VRRP without the limitations. VRRPE is unlike VRRP in the following ways:

- There is no "Owner" router. You do not need to use an IP address configured on one of the Layer 3 Switches as the virtual router ID (VRID), which is the address you are backing up for redundancy. The VRID is independent of the IP interfaces configured in the Layer 3 Switches. As a result, the protocol does not have an "Owner" as VRRP does.
- There is no restriction on which router can be the default master router. In VRRP, the "Owner" (the Layer 3 Switch on which the IP interface that is used for the VRID is configured) must be the default Master.

Foundry Layer 3 Switches configured for VRRPE can interoperate only with other Foundry Layer 3 Switches.

Architectural Differences

The protocols have the following architectural differences.

Management Protocol

- VRRP – VRRP routers send VRRP Hello and Hello messages to IP Multicast address 224.0.0.18.
- VRRPE – VRRPE sends messages to destination MAC address 01-00-5E-00-00-02 and destination IP address 224.0.0.2 (the standard IP multicast address for "all routers").

Virtual Router IP Address (the address you are backing up)

- VRRP – The virtual router IP address is the same as an IP address or virtual interface configured on one of the Layer 3 Switches, which is the "Owner" and becomes the default Master.
- VRRPE – The virtual router IP address is the gateway address you want to backup, but does not need to be an IP interface configured on one of the Layer 3 Switch's ports or a virtual interface.

Master and Backups

- VRRP – The "Owner" of the IP address of the VRID is the default Master and has the highest priority (255). The precedence of the Backups is determined by their priorities. The default Master is always the Owner of the IP address of the VRID.
- VRRPE – The Master and Backups are selected based on their priority. You can configure any of the Layer 3 Switches to be the Master by giving it the highest priority. There is no Owner.

VRRP and VRRPE Parameters

Table 37.1 lists the VRRP and VRRPE parameters. Most of the parameters and default values are the same for both protocols. The exceptions are noted in the table.

Table 37.1: VRRP and VRRPE Parameters

Parameter	Description	Default	See page...
Protocol	The Virtual Router Redundancy Protocol (VRRP) based on RFC 2338 or VRRP-Extended, Foundry's enhanced implementation of VRRP	Disabled **Note**: Only one of the protocols can be enabled at a time.	37-11 37-12
VRRP or VRRPE router	The Foundry Layer 3 Switch's active participation as a VRRP or VRRPE router. Enabling the protocol does not activate the Layer 3 Switch for VRRP or VRRPE. You must activate the device as a VRRP or VRRPE router after you configure the VRRP or VRRPE parameters.	Inactive	37-11 37-12
Virtual Router ID (VRID)	The ID of the virtual router you are creating by configuring multiple routers to back up an IP interface. You must configure the same VRID on each router that you want to use to back up the address. No default.	None	37-3 37-11 37-12
Virtual Router IP address	This is the address you are backing up. No default. • VRRP – The virtual router IP address must be a real IP address configured on the VRID interface on one of the VRRP routers. This router is the IP address Owner and is the default Master. • VRRPE – The virtual router IP address must be in the same subnet as a real IP address configured on the VRRPE interface, but cannot be the same as a real IP address configured on the interface.	None	37-3 37-11 37-12
VRID MAC address	The source MAC address in VRRP or VRRPE packets sent from the VRID interface, and the destination for packets sent to the VRID. • VRRP – A virtual MAC address defined as 00-00-5e-00-01-<vrid>. The Master owns the Virtual MAC address. • VRRPE – A virtual MAC address defined as 02-E0-52-<hash-value>-<vrid>, where <hash-value> is a two-octet hashed value for the IP address and <vrid> is the VRID.	Not configurable	37-3

Table 37.1: VRRP and VRRPE Parameters (Continued)

Parameter	Description	Default	See page...
Authentication type	The type of authentication the VRRP or VRRPE routers use to validate VRRP or VRRPE packets. The authentication type must match the authentication type the VRID's port uses with other routing protocols such as OSPF. • No authentication – The interfaces do not use authentication. This is the VRRP default. • Simple – The interface uses a simple text-string as a password in packets sent on the interface. If the interface uses simple password authentication, the VRID configured on the interface must use the same authentication type and the same password. **Note**: MD5 is not supported by VRRP or VRRPE.	No authentication	37-5 37-13
Router type	Whether the router is an Owner or a Backup. • Owner (VRRP only) – The router on which the real IP address used by the VRID is configured. • Backup – Routers that can provide routing services for the VRID but do not have a real IP address matching the VRID.	VRRP – The Owner is always the router that has the real IP address used by the VRID. All other routers for the VRID are Backups. VRRPE – All routers for the VRID are Backups.	37-14
Backup priority	A numeric value that determines a Backup's preferability for becoming the Master for the VRID. During negotiation, the router with the highest priority becomes the Master. • VRRP – The Owner has the highest priority (255); other routers can have a priority from 3 – 254. • VRRPE – All routers are Backups and have the same priority by default. If two or more Backups are tied with the highest priority, the Backup interface with the highest IP address becomes the Master for the VRID.	VRRP – 255 for the Owner; 100 for each Backup VRRPE – 100 for all Backups	37-14
Suppression of RIP advertisements	A router that is running RIP normally advertises routes to a backed up VRID even when the router is not currently the active router for the VRID. Suppression of these advertisements helps ensure that other routers do not receive invalid route paths for the VRID.	Disabled	37-15
Hello interval	The number of seconds between Hello messages from the Master to the Backups for a given VRID. The interval can from 1 – 84 seconds.	One second	37-4 37-16

Table 37.1: VRRP and VRRPE Parameters (Continued)

Parameter	Description	Default	See page...
Dead interval	The number of seconds a Backup waits for a Hello message from the Master for the VRID before determining that the Master is no longer active. If the Master does not send a Hello message before the dead interval expires, the Backups negotiate (compare priorities) to select a new Master for the VRID.	Three times the Hello Interval plus one-half second	37-4 37-16
Backup Hello interval	The number of seconds between Hello messages from a Backup to the Master. The message interval can be from 60 – 3600 seconds. You must enable the Backup to send the messages. The messages are disabled by default on Backups. The current Master (whether the VRRP Owner or a Backup) sends Hello messages by default.	Disabled 60 seconds when enabled	37-4 37-16
Track port	Another Layer 3 Switch port or virtual interface whose link status is tracked by the VRID's interface. If the link for a tracked interface goes down, the VRRP or VRRPE priority of the VRID interface is changed, causing the devices to renegotiate for Master.	None	37-4 37-17
Track priority	A VRRP or VRRPE priority value assigned to the tracked port(s). If a tracked port's link goes down, the VRID port's VRRP or VRRPE priority changes. • VRRP – The priority changes to the value of the tracked port's priority. • VRRPE – The VRID port's priority is reduced by the amount of the tracked port's priority.	VRRP – 2 VRRPE – 5	37-4 37-17
Backup preempt mode	Prevents a Backup with a higher VRRP priority from taking control of the VRID from another Backup that has a lower priority but has already assumed control of the VRID.	Enabled	37-17
Timer scale	Adjusts the timers for the Hello interval, Dead interval, Backup Hello interval, and Hold-down interval.	1	37-18
VRRP-E slow start timer	This feature causes a specified amount of time to elapse between the time the Master is restored and when it takes over from the Backup. This interval allows time for OSPF convergence when the Master is restored.	Disabled	37-18

Configuring Basic VRRP Parameters

To implement a simple VRRP configuration using all the default values, enter commands such as the following.

Configuring the Owner

```
Router1(config)#router vrrp
```

```
Router1(config)#inter e 1/6
Router1(config-if-1/6)#ip address 192.53.5.1
Router1(config-if-1/6)#ip vrrp vrid 1
Router1(config-if-1/6-vrid-1)#owner
Router1(config-if-1/6-vrid-1)#ip-address 192.53.5.1
Router1(config-if-1/6-vrid-1)#activate
```

Configuring a Backup

```
Router2(config)#router vrrp
Router2(config)#inter e 1/5
Router2(config-if-1/5)#ip address 192.53.5.3
Router2(config-if-1/5)#ip vrrp vrid 1
Router2(config-if-1/5-vrid-1)#backup
Router2(config-if-1/5-vrid-1)#advertise backup
Router2(config-if-1/5-vrid-1)#ip-address 192.53.5.1
Router2(config-if-1/5-vrid-1)#activate
```

Configuration Rules for VRRP

- The interfaces of all routers in a VRID must be in the same IP subnet.
- The IP addresses associated with the VRID must already be configured on the router that will be the Owner router.
- An IP address associated with the VRID must be on only one router.
- The Hello interval must be set to the same value on both the Owner and Backup(s) for the VRID.
- The Dead interval must be set to the same value on both the Owner and Backup(s) for the VRID.
- The track priority on a router must be lower than the router’s VRRP priority. Also, the track priority on the Owner must be higher than the track priority on the Backups.

Configuring Basic VRRPE Parameters

To implement a simple VRRPE configuration using all the default values, enter commands such as the following on each Layer 3 Switch.

```
Router2(config)#router vrrp-extended
Router2(config)#inter e 1/5
Router2(config-if-1/5)#ip address 192.53.5.3
Router2(config-if-1/5)#ip vrrp-extended vrid 1
Router2(config-if-1/5-vrid-1)#backup
Router2(config-if-1/5-vrid-1)#advertise backup
Router2(config-if-1/5-vrid-1)#ip-address 192.53.5.254
Router2(config-if-1/5-vrid-1)#activate
```

NOTE: You also can use the **enable** command to activate the configuration. This command does the same thing as the **activate** command.

Configuration Rules for VRRPE

- The interfaces of all routers in a VRID must be in the same IP subnet.
- The IP addresse) associated with the VRID cannot be configured on any of the Layer 3 Switches.
- The Hello interval must be set to the same value on all the Layer 3 Switches.
- The Dead interval must be set to the same value on all the Layer 3 Switches.
- The track priority for a VRID must be lower than the VRRPE priority.

Note Regarding Disabling VRRP or VRRPE

If you disable VRRP or VRRPE, the Layer 3 Switch removes all the configuration information for the disabled protocol from the running-config. Moreover, when you save the configuration to the startup-config file after disabling one of these protocols, all the configuration information for the disabled protocol is removed from the startup-config file.

The CLI displays a warning message such as the following:

```
Router1(config-vrrp-router)#no router vrrp
router vrrp mode now disabled. All vrrp config data will be lost when writing to
flash!
```

If you have disabled the protocol but have not yet saved the configuration to the startup-config file and reloaded the software, you can restore the configuration information by re-entering the command to enable the protocol (ex: **router vrrp**). If you have already saved the configuration to the startup-config file and reloaded the software, the information is gone.

If you are testing a VRRP or VRRPE configuration and are likely to disable and re-enable the protocol, you might want to make a backup copy of the startup-config file containing the protocol's configuration information. This way, if you remove the configuration information by saving the configuration after disabling the protocol, you can restore the configuration by copying the backup copy of the startup-config file onto the flash memory.

Configuring Additional VRRP and VRRPE Parameters

You can modify the following VRRP and VRRPE parameters on an individual VRID basis. These parameters apply to both protocols:

- Authentication type (if the interfaces on which you configure the VRID use authentication)
- Router type (Owner or Backup)

NOTE: For VRRP, change the router type only if you have moved the real IP address from one router to another or you accidentally configured the IP address Owner as a Backup.

For VRRPE, the router type is always Backup. You cannot change the type to Owner.

- Backup priority
- Suppression of RIP advertisements on Backup routes for the backed up interface
- Hello interval
- Dead interval
- Backup Hello messages and message timer (Backup advertisement)
- Track port
- Track priority
- Backup preempt mode
- Timer scale
- VRRP-E slow start timer

For information about the fields, see the parameter descriptions in the following sections.

See “VRRP and VRRPE Parameters” on page 37-9 for a summary of the parameters and their defaults.

Authentication Type

If the interfaces on which you configure the VRID use authentication, the VRRP or VRRPE packets on those interfaces also must use the same authentication. Foundry's implementation of VRRP and VRRPE supports the following authentication types:

- No authentication – The interfaces do not use authentication. This is the default for VRRP and VRRPE.
- Simple – The interfaces use a simple text-string as a password in packets sent on the interface. If the interfaces use simple password authentication, the VRID configured on the interfaces must use the same authentication type and the same password.

To configure the VRID interface on Router1 for simple-password authentication using the password "ourpword", enter the following commands:

Configuring Router 1

```
Router1(config)#inter e 1/6
Router1(config-if-1/6)#ip vrrp auth-type simple-text-auth ourpword
```

Configuring Router 2

```
Router2(config)#inter e 1/5
Router2(config-if-1/5)#ip vrrp auth-type simple-text-auth ourpword
```

VRRP Syntax

Syntax: ip vrrp auth-type no-auth | simple-text-auth <auth-data>

The **auth-type no-auth** parameter indicates that the VRID and the interface it is configured on do not use authentication.

The **auth-type simple-text-auth** <auth-data> parameter indicates that the VRID and the interface it is configured on use a simple text password for authentication. The <auth-data> parameter is the password. If you use this parameter, make sure all interfaces on all the routers supporting this VRID are configured for simple password authentication and use the same password.

VRRPE Syntax

Syntax: ip vrrp-extended auth-type no-auth | simple-text-auth <auth-data>

The parameter values are the same as for VRRP.

Router Type

A VRRP interface is either an Owner or a Backup for a given VRID. By default, the Owner becomes the Master following the negotiation. A Backup becomes the Master only if the Master becomes unavailable.

A VRRPE interface is always a Backup for its VRID. The Backup with the highest VRRP priority becomes the Master.

This section describes how to specify the interface type, how to change the type for VRRP, and how to set or change the interface's VRRP or VRRPE priority and track priority for the VRID.

NOTE: You can force a VRRP master router to abdicate (give away control) of the VRID to a Backup by temporarily changing the Master's VRRP priority to a value less than the Backup's. See "Forcing a Master Router to Abdicate to a Standby Router" on page 37-19.

NOTE: The type Owner is not applicable to VRRPE.

NOTE: The IP address(es) you associate with the Owner must be a real IP address (or addresses) on the interface on which you configure the VRID.

When you configure a Backup router, the router interface on which you are configuring the VRID must have a real IP address that is in the same subnet as the address associated with the VRID by the Owner. However, the address cannot be the same.

To configure Router1 as a VRRP VRID's Owner, enter the following commands:

```
Router1(config)#inter e 1/6
Router1(config-if-1/6)#ip vrrp vrid 1
Router1(config-if-1/6-vrid-1)#owner
```

To configure Router2 as a VRRP Backup for the same VRID, enter the following commands:

```
Router2(config)#inter e 1/5
Router2(config-if-1/5)#ip vrrp vrid 1
Router2(config-if-1/5-vrid-1)#backup
Router2(config-if-1/5-vrid-1)#advertise backup
```

To configure a VRRPE interface as a Backup for a VRID and set its VRRPE priority and track priority, enter commands such as the following:

```
FastIron(config)#inter e 1/1
FastIron(config-if-1/1)#ip vrrp-extended vrid 1
FastIron(config-if-1/1-vrid-1)#backup priority 50 track-priority 10
Router2(config-if-1/1-vrid-1)#advertise backup
```

VRRP Syntax

Syntax: owner [track-priority <value>]

The **track-priority** <value> parameter changes the track-port priority for this interface and VRID from the default (2) to a value from 1 – 254.

Syntax: backup [priority <value>] [track-priority <value>]

The **priority** <value> parameter specifies the VRRP priority for this interface and VRID. You can specify a value from 3 – 254. The default is 100.

The **track-priority** <value> parameter is the same as above.

NOTE: You cannot set the priority of a VRRP Owner. The Owner's priority is always 255.

VRRPE Syntax

Syntax: backup [priority <value>] [track-priority <value>]

The software requires you to identify a VRRPE interface as a Backup for its VRID before you can activate the interface for the VRID. However, after you configure the VRID, you can use this command to change its priority or track priority. The parameter values are the same as for VRRP.

Suppression of RIP Advertisements on Backup Routers for the Backup Interface

Normally, a VRRP or VRRPE Backup includes route information for the virtual IP address (the backed up interface) in RIP advertisements. As a result, other routers receive multiple paths for the backed up interface and might sometimes unsuccessfully use the path to the Backup rather than the path to the Master.

You can prevent the Backups from advertising route information for the backed up interface by enabling suppression of the advertisements.

To suppress RIP advertisements for the backed up interface in Router2, enter the following commands:

```
Router2(config)#router rip
Router2(config-rip-router)#use-vrrp-path
```

Syntax: use-vrrp-path

The syntax is the same for VRRP and VRRPE.

Hello Interval

The Master periodically sends Hello messages to the Backups. The Backups use the Hello messages as verification that the Master is still on-line. If the Backup routers stop receiving the Hello messages for the period of time specified by the Dead interval, the Backup routers determine that the Master router is dead. At this point, the Backup router with the highest priority becomes the new Master router. The Hello interval can be from 1 – 84 seconds. The default is 1 second.

NOTE: The default Dead interval is three times the Hello Interval plus one-half second. Generally, if you change the Hello interval, you also should change the Dead interval on the Backup routers.

To change the Hello interval on the Master to 10 seconds, enter the following commands:

```
Router1(config)#inter e 1/6
Router1(config-if-1/6)#ip vrrp vrid 1
Router1(config-if-1/6-vrid-1)#hello-interval 10
```

Syntax: hello-interval <value>

The syntax is the same for VRRP and VRRPE.

Dead Interval

The Dead interval is the number of seconds a Backup waits for a Hello message from the Master before determining that the Master is dead. When Backups determine that the Master is dead, the Backup with the highest priority becomes the new Master. The Dead interval can be from 1 – 84 seconds. The default is 3.5 seconds. This is three times the default Hello interval (1 second) plus one-half second added by the router software. The software automatically adds one-half second to the Dead interval value you enter.

To change the Dead interval on a Backup to 30 seconds, enter the following commands:

```
Router2(config)#inter e 1/5
Router2(config-if-1/5)#ip vrrp vrid 1
Router2(config-if-1/5-vrid-1)#dead-interval 30
```

Syntax: dead-interval <value>

The syntax is the same for VRRP and VRRPE.

Backup Hello Message State and Interval

By default, Backup do not send Hello messages to advertise themselves to the Master. You can enable these messages if desired and also change the message interval.

To enable a Backup to send Hello messages to the Master, enter commands such as the following:

```
FastIron(config)#router vrrp
FastIron(config)#inter e 1/6
FastIron(config-if-1/6)#ip vrrp vrid 1
FastIron(config-if-1/6-vrid-1)#advertise backup
```

Syntax: [no] advertise backup

When you enable a Backup to send Hello messages, the Backup sends a Hello messages to the Master every 60 seconds by default. You can change the interval to be up to 3600 seconds. To do so, enter commands such as the following:

```
FastIron(config)#router vrrp
FastIron(config)#inter e 1/6
FastIron(config-if-1/6)#ip vrrp vrid 1
FastIron(config-if-1/6-vrid-1)#backup-hello-interval 180
```

Syntax: [no] backup-hello-interval <num>

The <num> parameter specifies the message interval and can be from 60 – 3600 seconds. The default is 60 seconds.

The syntax is the same for VRRP and VRRPE.

Track Port

You can configure the VRID on one interface to track the link state of another interface on the Layer 3 Switch. This capability is quite useful for tracking the state of the exit interface for the path for which the VRID is providing redundancy. See “Track Ports and Track Priority” on page 37-4.

To configure 1/6 on Router1 to track interface 2/4, enter the following commands:

```
Router1(config)#inter e 1/6
Router1(config-if-1/6)#ip vrrp vrid 1
Router1(config-if-1/6-vrid-1)#track-port e 2/4
```

Syntax: track-port ethernet [<slotnum>/]<portnum> | ve <num>

The syntax is the same for VRRP and VRRPE.

Track Priority

When you configure a VRID to track the link state of other interfaces, if one of the tracked interface goes down, the software changes the VRRP or VRRPE priority of the VRID interface.

- For VRRP, the software changes the priority of the VRID to the track priority, which typically is lower than the VRID priority and lower than the VRID’s priorities configured on the Backups. For example, if the VRRPE interface’s priority is 100 and a tracked interface with track priority 60 goes down, the software changes the VRRPE interface’s priority to 60.
- For VRRPE, the software reduces the VRID priority by the amount of the priority of the tracked interface that went down. For example, if the VRRPE interface’s priority is 100 and a tracked interface with track priority 60 goes down, the software changes the VRRPE interface’s priority to 40. If another tracked interface goes down, the software reduces the VRID’s priority again, by the amount of the tracked interface’s track priority.

The default track priority for a VRRP Owner is 2. The default track priority for Backups is 1.

You enter the track priority as a parameter with the **owner** or **backup** command. See “Track Port” on page 37-17.

Syntax: owner [track-priority <value>]

Syntax: backup [priority <value>] [track-priority <value>]

The syntax is the same for VRRP and VRRPE.

Backup Preempt

By default, a Backup that has a higher priority than another Backup that has become the Master can preempt the Master, and take over the role of Master. If you want to prevent this behavior, disable preemption.

Preemption applies only to Backups and takes effect only when the Master has failed and a Backup has assumed ownership of the VRID. The feature prevents a Backup with a higher priority from taking over as Master from another Backup that has a lower priority but has already become the Master of the VRID.

Preemption is especially useful for preventing flapping in situations where there are multiple Backups and a Backup with a lower priority than another Backup has assumed ownership, because the Backup with the higher priority was unavailable when ownership changed.

If you enable the non-preempt mode (thus disabling the preemption feature) on all the Backups, the Backup that becomes the Master following the disappearance of the Master continues to be the Master. The new Master is not preempted.

NOTE: In VRRP, regardless of the setting for the preempt parameter, the Owner always becomes the Master again when it comes back online.

To disable preemption on a Backup, enter commands such as the following:

```
Router1(config)#inter e 1/6
Router1(config-if-1/6)#ip vrrp vrid 1
Router1(config-if-1/6-vrid-1)#non-preempt-mode
```

Syntax: non-preempt-mode

The syntax is the same for VRRP and VRRPE.

Changing the Timer Scale

Platform Support:

- FESX and FSX devices running software release 03.0.00 and later

To achieve sub-second failover times, you can shorten the duration of all scale timers for VSRP, VRRP, and VRRP-E by adjusting the timer scale. The ***timer scale*** is a value used by the software to calculate the timers. By default, the scale value is 1. If you increase the timer scale, each timer's value is divided by the scale value. Using the timer scale to adjust timer values enables you to easily change all the timers while preserving the ratios among their values. Here is an example.

Timer	Timer Scale	Timer Value
Hello interval	1	1 second
	2	0.5 seconds
Dead interval	1	3 seconds
	2	1.5 seconds
Backup Hello interval	1	60 seconds
	2	30 seconds
Hold-down interval	1	2 seconds
	2	1 second

If you configure the device to receive its timer values from the Master, the Backup also receives the timer scale value from the Master.

NOTE: The Backups always use the value of the timer scale received from the Master, regardless of whether the timer values that are saved in the configuration are the values configured on the Backup or the values received from the Master.

To change the timer scale, enter a command such as the following at the global CONFIG level of the CLI:

```
FastIron SuperX Router(config)# scale-timer 2
```

This command changes the scale to 2. All VSRP, VRRP, and VRRP-E timer values will be divided by 2.

Syntax: [no] scale-timer <num>

The <num> parameter specifies the multiplier. You can specify a timer scale from 1 – 10.

VRRP-E Slow Start Timer

Platform Support:

- FESX and FSX devices running software release 03.0.00 and later

In a VRRP-E configuration, if a Master router goes down, the Backup router with the highest priority takes over. When the Master comes back up again, it takes over from the Backup. By default, this transition from Backup back to Master takes place immediately. However, you can configure the VRRP-E slow start timer feature, which causes a specified amount of time to elapse between the time the Master is restored and when it takes over from the Backup. This interval allows time for OSPF convergence when the Master is restored.

NOTE: This feature is not supported in the FGS.

To set the VRRP-E slow start timer to 30 seconds, enter the following commands:

```
FastIron(config)#router vrrp-e
FastIron(config-vrrpe-router)#slow-start 30
```

Syntax: [no] slow-start <seconds>

For <seconds>, enter a value from 1 – 255.

When the VRRP-E slow start timer is enabled, if the Master goes down, the Backup takes over immediately. If the Master subsequently comes back up again, the amount of time specified by the VRRP-E slow start timer elapses (in this example, 30 seconds) before the Master takes over from the Backup.

The VRRP-E slow start timer is effective only if another VRRP-E Master (Standby) is detected. It is not effective during the initial boot up.

NOTE: The VRRP-E slow start timer applies only to VRRP-E configurations. It does not apply to VRRP configurations.

Forcing a Master Router to Abdicate to a Standby Router

You can force a VRRP Master to abdicate (give away control) of a VRID to a Backup by temporarily changing the Master's priority to a value less than the Backup's.

The VRRP Owner always has priority 255. You can even use this feature to temporarily change the Owner's priority to a value from 1 – 254.

NOTE: When you change a VRRP Owner's priority, the change takes effect only for the current power cycle. The change is not saved to the startup-config file when you save the configuration and is not retained across a reload or reboot. Following a reload or reboot, the VRRP Owner again has priority 255.

To temporarily change the Master's priority, use the following CLI method.

To change the Master's priority, enter commands such as the following:

```
FastIron(config)#ip int eth 1/6
FastIron(config-if-1/6)#ip vrrp vrid 1
FastIron(config-if-1/6-vrid-1)#owner priority 99
```

Syntax: [no] owner priority | track-priority <num>

The <num> parameter specifies the new priority and can be a number from 1 – 254.

When you press Enter, the software changes the priority of the Master to the specified priority. If the new priority is lower than at least one Backup's priority for the same VRID, the Backup takes over and becomes the new Master until the next software reload or system reset.

To verify the change, enter the following command from any level of the CLI:

```
FastIron#show ip vrrp
Total number of VRRP routers defined: 1
Interface ethernet 1/6
auth-type no authentication
```

```
VRID 1
state backup
administrative-status enabled
mode owner
priority 99
current priority 99
hello-interval 1 sec
ip-address 192.53.5.1
backup routers 192.53.5.2
```

This example shows that even though this Layer 3 Switch is the Owner of the VRID (“mode owner”), the Layer 3 Switch’s priority for the VRID is only 99 and the state is now “backup” instead of “active”. In addition, the administrative status is “enabled”.

To change the Master’s priority back to the default Owner priority 255, enter “no” followed by the command you entered to change the priority. For example, to change the priority of a VRRP Owner back to 255 from 99, enter the following command:

```
FastIron(config-if-1/6-vrid-1)#no owner priority 99
```

You cannot set the priority to 255 using the **owner priority** command.

Displaying VRRP and VRRPE Information

You can display the following information for VRRP or VRRPE:

- Summary configuration and status information
- Detailed configuration and status information
- VRRP and VRRPE Statistics
- CPU utilization statistics

Displaying Summary Information

To display summary information for a Layer 3 Switch, enter the following command at any level of the CLI:

```
FastIron#show ip vrrp brief

Total number of VRRP routers defined: 1
Interface VRID CurPri P State  Master addr   Backup addr       VIP
 1/6        1    255  P Init   192.53.5.1    192.53.5.3 192.53.5.1
```

The above example is for VRRP. Here is an example for VRRPE:

```
FastIron#show ip vrrp-extended brief

Total number of VRRP-Extended routers defined: 1
Interface VRID CurPri P State  Master addr   Backup addr       VIP
 1/6        1    255  P Init   192.53.5.2    192.53.5.3 192.53.5.254
```

Syntax: show ip vrrp brief | ethernet [<slotnum>/]<portnum> | ve <num> | stat

Syntax: show ip vrrp-extended brief | ethernet [<slotnum>/]<portnum> | ve <num> | stat

The **brief** parameter displays the summary information. If you do not use this parameter, detailed information is displayed instead. See “Displaying Detailed Information” on page 37-22.

The <slotnum> parameter is required on chassis devices if you specify a port number.

The <portnum> parameter specifies an Ethernet port. If you use this parameter, the command displays VRRP or VRRPE information only for the specified port.

The **ve** <num> parameter specifies a virtual interface. If you use this parameter, the command displays VRRP or VRRPE information only for the specified virtual interface.

The **stat** parameter displays statistics. See “Displaying Statistics” on page 37-28.

This display shows the following information.

Table 37.2: CLI Display of VRRP or VRRPE Summary Information

This Field...	Displays...
Total number of VRRP (or VRRP-Extended) routers defined	The total number of VRIDs configured on this Layer 3 Switch. **Note**: The total applies only to the protocol the Layer 3 Switch is running. For example, if the Layer 3 Switch is running VRRPE, the total applies only to VRRPE routers.
Interface	The interface on which VRRP or VRRPE is configured. If VRRP or VRRPE is configured on multiple interfaces, information for each interface is listed separately.
VRID	The VRID configured on this interface. If multiple VRIDs are configured on the interface, information for each VRID is listed in a separate row.
CurPri	The current VRRP or VRRPE priority of this Layer 3 Switch for the VRID.
P	Whether the backup preempt mode is enabled. If the backup preempt mode is enabled, this field contains a “P”. If the mode is disabled, this field is blank.
State	This Layer 3 Switch’s VRRP or VRRPE state for the VRID. The state can be one of the following: • Init – The VRID is not enabled (activated). If the state remains Init after you activate the VRID, make sure that the VRID is also configured on the other routers and that the routers can communicate with each other. **Note**: If the state is Init and the mode is incomplete, make sure you have specified the IP address for the VRID. • Backup – This Layer 3 Switch is a Backup for the VRID. • Master – This Layer 3 Switch is the Master for the VRID.
Master addr	IP address of the router interface that is currently Master for the VRID.
Backup addr	IP addresses of router interfaces that are currently Backups for the VRID.
VIP	The virtual IP address that is being backed up by the VRID.

Displaying Detailed Information

To display detailed VRRP or VRRPE information, enter the following command at any level of the CLI:

```
FastIron#show ip vrrp

Total number of VRRP routers defined: 1
Interface ethernet 1/6
 auth-type no authentication
 VRID 1
  state master
  administrative-status enabled
  mode owner
  priority 255
  current priority 255
  hello-interval 10000 msec
  advertise backup: disabled
  track-port 2/4
```

This example is for a VRRP Owner. Here is an example for a VRRP Backup.

```
FastIron#show ip vrrp

Total number of VRRP routers defined: 1
Interface ethernet 1/5
 auth-type no authentication
 VRID 1
  state backup
  administrative-status enabled
  mode non-owner(backup)
  priority 100
  current priority 100
  hello-interval 10000 msec
  dead-interval 30000 msec
  current dead-interval 10000 msec
  preempt-mode true
  advertise backup: enabled
  backup router 192.53.5.3 expires in 00:00:03.0
  next hello sent in 00:00:02.0
  track-port 3/2
```

Here is an example for a VRRPE Backup.

```
FastIron#show ip vrrp-extended

Total number of VRRP-Extended routers defined: 1
Interface ethernet 1/6
 auth-type no authentication
 VRID 1
  state master
  administrative-status enabled
  priority 200
  current priority 200
  hello-interval 10000 msec
  dead-interval 30000 msec
  current dead-interval 30000 msec
  preempt-mode true
  virtual ip address 192.53.5.254
  advertise backup: enabled
  master router 192.53.5.2 expires in 00:00:03.0
  track-port 2/4
```

Syntax: show ip vrrp brief | ethernet [<slotnum>/]<portnum> | ve <num> | stat

Syntax: show ip vrrp-extended brief | ethernet [<slotnum>/]<portnum> | ve <num> | stat

The **brief** parameter displays summary information. See “Displaying Summary Information” on page 37-20.

The <portnum> parameter specifies an Ethernet port. If you use this parameter, the command displays VRRP or VRRPE information only for the specified port. Also, you must specify the <slotnum> on chassis devices.

The **ve** <num> parameter specifies a virtual interface. If you use this parameter, the command displays VRRP or VRRPE information only for the specified virtual interface.

The **stat** parameter displays statistics. See “Displaying Statistics” on page 37-28.

This display shows the following information.

Table 37.3: CLI Display of VRRP or VRRPE Detailed Information

This Field...	Displays...
Total number of VRRP (or VRRP-Extended) routers defined	The total number of VRIDs configured on this Layer 3 Switch. **Note**: The total applies only to the protocol the Layer 3 Switch is running. For example, if the Layer 3 Switch is running VRRPE, the total applies only to VRRPE routers.
Interface parameters	
Interface	The interface on which VRRP or VRRPE is configured. If VRRP or VRRPE is configured on multiple interfaces, information for each interface is listed separately.
auth-type	The authentication type enabled on the interface.
VRID parameters	
VRID	The VRID configured on this interface. If multiple VRIDs are configured on the interface, information for each VRID is listed separately.

Table 37.3: CLI Display of VRRP or VRRPE Detailed Information (Continued)

This Field...	Displays...
state	This Layer 3 Switch's VRRP or VRRPE state for the VRID. The state can be one of the following: • initialize – The VRID is not enabled (activated). If the state remains "initialize" after you activate the VRID, make sure that the VRID is also configured on the other routers and that the routers can communicate with each other. **Note**: If the state is "initialize" and the mode is incomplete, make sure you have specified the IP address for the VRID. • backup – This Layer 3 Switch is a Backup for the VRID. • master – This Layer 3 Switch is the Master for the VRID.
administrative-status	The administrative status of the VRID. The administrative status can be one of the following: • disabled – The VRID is configured on the interface but VRRP or VRRPE has not been activated on the interface. • enabled – VRRP or VRRPE has been activated on the interface.
mode	Indicates whether the Layer 3 Switch is the Owner or a Backup for the VRID. **Note**: If "incomplete" appears after the mode, configuration for this VRID is incomplete. For example, you might not have configured the virtual IP address that is being backed up by the VRID. **Note**: This field applies only to VRRP. All Layer 3 Switches configured for VRRPE are Backups.
priority	The device's preferability for becoming the Master for the VRID. During negotiation, the router with the highest priority becomes the Master. If two or more devices are tied with the highest priority, the Backup interface with the highest IP address becomes the active router for the VRID.
current priority	The current VRRP or VRRPE priority of this Layer 3 Switch for the VRID. The current priority can differ from the configured priority (see the row above) for the following reasons: • The VRID is still in the initialization stage and has not become a Master or Backup yet. In this case, the current priority is 0. • The VRID is configured with track ports and the link on a tracked interface has gone down. See "Track Ports and Track Priority" on page 37-4.
hello-interval	The configured value for the hello interval. This is the amount of time between Hello messages from the Master to the Backups for a given VRID. • Software releases prior to 03.0.00 show the hello interval in number of *seconds*. • Software releases 03.0.00 and later show the hello interval in number of *milliseconds*.

Table 37.3: CLI Display of VRRP or VRRPE Detailed Information (Continued)

This Field...	Displays...
dead-interval	The configured value for the dead interval. This is the amount of time a Backup waits for a Hello message from the Master for the VRID before determining that the Master is no longer active. • Software releases prior to 03.0.00 show the dead interval in number of *seconds*. • Software releases 03.0.00 and later show the dead interval in number of *milliseconds*. If the Master does not send a Hello message before the dead interval expires, the Backups negotiate (compare priorities) to select a new Master for the VRID. **Note**: If the value is 0, then you have not configured this parameter. **Note**: This field does not apply to VRRP Owners.
current dead-interval	The current value of the dead interval. This is the value actually in use by this interface for the VRID. • Software releases prior to 03.0.00 show the current dead interval in number of *seconds*. • Software releases 03.0.00 and later show the current dead interval in number of *milliseconds*. **Note**: This field does not apply to VRRP Owners.
preempt-mode	Whether the backup preempt mode is enabled. **Note**: This field does not apply to VRRP Owners.
virtual ip address	The virtual IP addresses that this VRID is backing up.
advertise backup	The IP addresses of Backups that have advertised themselves to this Layer 3 Switch by sending Hello messages. **Note**: Hello messages from Backups are disabled by default. You must enable the Hello messages on the Backup for the Backup to advertise itself to the current Master. See “Hello Messages” on page 37-4.
backup router <ip-addr> expires in <time>	The IP addresses of Backups that have advertised themselves to this Master by sending Hello messages. The <time> value indicates how long before the Backup expires. A Backup expires if you disable the advertise backup option on the Backup or the Backup becomes unavailable. Otherwise, the Backup’s next Hello message arrives before the Backup expires. The Hello message resets the expiration timer. An expired Backup does not necessarily affect the Master. However, if you have not disabled the advertise backup option on the Backup, then the expiration may indicate a problem with the Backup. **Note**: This field applies only when Hello messages are enabled on the Backups (using the advertise backup option).

Table 37.3: CLI Display of VRRP or VRRPE Detailed Information (Continued)

This Field...	Displays...
next hello sent in <time>	How long until the Backup sends its next Hello message. **Note**: This field applies only when this Layer 3 Switch is the Master and the Backup is configured to send Hello messages (the advertise backup option is enabled).
master router <ip-addr> expires in <time>	The IP address of the Master and the amount of time until the Master's dead interval expires. If the Backup does not receive a Hello message from the Master by the time the interval expires, either the IP address listed for the Master will change to the IP address of the new Master, or this Layer 3 Switch itself will become the Master. **Note**: This field applies only when this Layer 3 Switch is a Backup.
track port	The interfaces that the VRID's interface is tracking. If the link for a tracked interface goes down, the VRRP or VRRPE priority of the VRID interface is changed, causing the devices to renegotiate for Master. **Note**: This field is displayed only if track interfaces are configured for this VRID.

Displaying Detailed Information for an Individual VRID

You can display information about the settings configured for a specified VRRP Virtual Router ID (VRID). For example, to display information about VRID 1:

```
FastIron#show ip vrrp vrid 1
VRID 1
  Interface ethernet 3/11
  state initialize
  administrative-status disabled
  mode non-owner(backup)incomplete
  priority 12
  current priority 12
  track-priority 22
  hello-interval 1 sec
  dead-interval 0 sec
  current dead-interval 3.900 sec
  preempt-mode true
  advertise backup: disabled
```

Syntax: show ip vrrp vrid <num> [ethernet <num> | ve <num>]

The <num> parameter specifies the VRID.

The **ethernet** <num> | **ve** <num> specifies an interface on which the VRID is configured. If you specify an interface, VRID information is displayed for that interface only. Otherwise, information is displayed for all the interfaces on which the specified VRID is configured.

This display shows the following information.

Table 37.4: Output from the show ip vrrp vrid command

This Field...	Displays...
VRID	The specified VRID.
Interface	The interface on which VRRP is configured.
State	This Layer 3 Switch's VRRP state for the VRID. The state can be one of the following: • Init – The VRID is not enabled (activated). If the state remains Init after you activate the VRID, make sure that the VRID is also configured on the other routers and that the routers can communicate with each other. **Note**: If the state is Init and the mode is incomplete, make sure you have specified the IP address for the VRID. • Backup – This Layer 3 Switch is a Backup for the VRID. • Master – This Layer 3 Switch is the Master for the VRID.
priority	The configured VRRP priority of this Layer 3 Switch for the VRID.
current priority	The current VRRP priority of this Layer 3 Switch for the VRID.
track-priority	The new VRRP priority that the router receives for this VRID if the interface goes down
hello-interval	How often the Master router sends Hello messages to the Backups.
dead-interval	The amount of time a Backup waits for a Hello message from the Master before determining that the Master is dead.
current dead-interval	The current Dead interval. The software automatically adds one-half second to the Dead interval value you enter.
preempt-mode	Whether the backup preempt mode is enabled. If the backup preempt mode is enabled, this field contains "true". If the mode is disabled, this field contains "false".
advertise backup	Whether Backup routers send Hello messages to the Master.

Displaying Statistics

To display statistics on most Foundry devices, enter a command such as the following at any level of the CLI:

```
FastIron#show ip vrrp statistic

Interface ethernet 1/5
 rxed vrrp header error count = 0
 rxed vrrp auth error count = 0
 rxed vrrp auth passwd mismatch error count = 0
 rxed vrrp vrid not found error count = 0
 VRID 1
 rxed arp packet drop count = 0
 rxed ip packet drop count = 0
 rxed vrrp port mismatch count = 0
 rxed vrrp ip address mismatch count = 0
 rxed vrrp hello interval mismatch count = 0
 rxed vrrp priority zero from master count = 0
 rxed vrrp higher priority count = 0
 transitioned to master state count = 1
 transitioned to backup state count = 1
```

The same statistics are listed for VRRP and VRRPE.

Syntax: show ip vrrp brief | ethernet [<slotnum>/]<portnum> | ve <num> | statistic

Syntax: show ip vrrp-extended brief | ethernet [<slotnum>/]<portnum> | ve <num> | stat

The **brief** parameter displays summary information. See “Displaying Summary Information” on page 37-20.

If you specify a port, the <slotnum> parameter is required on chassis devices.

The <portnum> parameter specifies an Ethernet port. If you use this parameter, the command displays detailed VRRP or VRRPE information only for the specified port. See “Displaying Detailed Information” on page 37-22.

The **ve** <num> parameter specifies a virtual interface. If you use this parameter, the command displays detailed VRRP or VRRPE information only for the specified virtual interface. See “Displaying Detailed Information” on page 37-22.

The **statistic** parameter displays statistics. This parameter is required for displaying the statistics.

This display shows the following information.

Table 37.5: CLI Display of VRRP or VRRPE Statistics

This Field...	Displays...
Interface Statistics	
Interface	The interface on which VRRP or VRRPE is configured. If VRRP or VRRPE is configured on more than one interface, the display lists the statistics separately for each interface.
rxed vrrp header error count	The number of VRRP or VRRPE packets received by the interface that had a header error.
rxed vrrp auth error count	The number of VRRP or VRRPE packets received by the interface that had an authentication error.

Table 37.5: CLI Display of VRRP or VRRPE Statistics (Continued)

This Field...	Displays...
rxed vrrp auth passwd mismatch error count	The number of VRRP or VRRPE packets received by the interface that had a password value that does not match the password used by the interface for authentication.
rxed vrrp vrid not found error count	The number of VRRP or VRRPE packets received by the interface that contained a VRID that is not configured on this interface.
VRID Statistics	
rxed arp packet drop count	The number of ARP packets addressed to the VRID that were dropped.
rxed ip packet drop count	The number of IP packets addressed to the VRID that were dropped.
rxed vrrp port mismatch count	The number of packets received that did not match the configuration for the receiving interface.
rxed vrrp ip address mismatch count	The number of packets received that did not match the configured IP addresses.
rxed vrrp hello interval mismatch count	The number of packets received that did not match the configured Hello interval.
rxed vrrp priority zero from master count	The current Master has resigned.
rxed vrrp higher priority count	The number of VRRP or VRRPE packets received by the interface that had a higher backup priority for the VRID than this Layer 3 Switch's backup priority for the VRID.
transitioned to master state count	The number of times this Layer 3 Switch has changed from the backup state to the master state for the VRID.
transitioned to backup state count	The number of times this Layer 3 Switch has changed from the master state to the backup state for the VRID.

Clearing VRRP or VRRPE Statistics

Use the following methods to clear VRRP or VRRPE statistics.

To clear VRRP or VRRPE statistics, enter the following command at the Privileged EXEC level or any configuration level of the CLI:

```
Router1#clear ip vrrp-stat
```

Syntax: clear ip vrrp-stat

Displaying CPU Utilization Statistics

You can display CPU utilization statistics for VRRP and other IP protocols.

To display CPU utilization statistics for the previous one-second, one-minute, five-minute, and fifteen-minute intervals, enter the following command at any level of the CLI:

```
FastIron#show process cpu
Process Name   5Sec(%)   1Min(%)   5Min(%)   15Min(%)   Runtime(ms)
ARP              0.01      0.03      0.09      0.22               9
BGP              0.04      0.06      0.08      0.14               13
GVRP             0.00      0.00      0.00      0.00               0
ICMP             0.00      0.00      0.00      0.00               0
IP               0.00      0.00      0.00      0.00               0
OSPF             0.00      0.00      0.00      0.00               0
RIP              0.00      0.00      0.00      0.00               0
STP              0.00      0.00      0.00      0.00               0
VRRP             0.03      0.07      0.09      0.10               8
```

If the software has been running less than 15 minutes (the maximum interval for utilization statistics), the command indicates how long the software has been running. Here is an example:

```
FastIron#show process cpu
The system has only been up for 6 seconds.
Process Name   5Sec(%)   1Min(%)   5Min(%)   15Min(%)   Runtime(ms)
ARP              0.01      0.00      0.00      0.00               0
BGP              0.00      0.00      0.00      0.00               0
GVRP             0.00      0.00      0.00      0.00               0
ICMP             0.01      0.00      0.00      0.00               1
IP               0.00      0.00      0.00      0.00               0
OSPF             0.00      0.00      0.00      0.00               0
RIP              0.00      0.00      0.00      0.00               0
STP              0.00      0.00      0.00      0.00               0
VRRP             0.00      0.00      0.00      0.00               0
```

To display utilization statistics for a specific number of seconds, enter a command such as the following:

```
FastIron#show process cpu 2
Statistics for last 1 sec and 80 ms
Process Name   Sec(%)   Time(ms)
ARP              0.00       0
BGP              0.00       0
GVRP             0.00       0
ICMP             0.01       1
IP               0.00       0
OSPF             0.00       0
RIP              0.00       0
STP              0.01       0
VRRP             0.00       0
```

When you specify how many seconds' worth of statistics you want to display, the software selects the sample that most closely matches the number of seconds you specified. In this example, statistics are requested for the previous two seconds. The closest sample available is actually for the previous 1 second plus 80 milliseconds.

Syntax: show process cpu [<num>]

The <num> parameter specifies the number of seconds and can be from 1 – 900. If you use this parameter, the command lists the usage statistics only for the specified number of seconds. If you do not use this parameter, the command lists the usage statistics for the previous one-second, one-minute, five-minute, and fifteen-minute intervals.

Configuration Examples

The following sections contain the CLI commands for implementing the VRRP and VRRPE configurations shown in Figure 37.2 on page 37-2 and Figure 37.3 on page 37-7.

VRRP Example

To implement the VRRP configuration shown in Figure 37.2 on page 37-2, use the following method.

Configuring Router1

To configure VRRP Router1, enter the following commands:

```
Router1(config)#router vrrp
Router1(config)#inter e 1/6
Router1(config-if-1/6)#ip address 192.53.5.1
Router1(config-if-1/6)#ip vrrp vrid 1
Router1(config-if-1/6-vrid-1)#owner track-priority 20
Router1(config-if-1/6-vrid-1)#track-port ethernet 2/4
Router1(config-if-1/6-vrid-1)#ip-address 192.53.5.1
Router1(config-if-1/6-vrid-1)#activate
```

NOTE: When you configure the Master (Owner), the address you enter with the **ip-address** command must already be configured on the interface.

The **ip vrrp owner** command specifies that this router owns the IP address you are associating with the VRID. Because this router owns the IP address, this router is the default Master router and its VRRP priority is thus 255.

Configuring Router2

To configure Router2 in Figure 37.2 on page 37-2 after enabling VRRP, enter the following commands:

```
Router2(config)#router vrrp
Router2(config)#inter e 1/5
Router2(config-if-1/5)#ip address 192.53.5.3
Router2(config-if-1/5)#ip vrrp vrid 1
Router2(config-if-1/5-vrid-1)#backup priority 100 track-priority 19
Router2(config-if-1/5-vrid-1)#track-port ethernet 3/2
Router2(config-if-1/5-vrid-1)#ip-address 192.53.5.1
Router2(config-if-1/5-vrid-1)#activate
```

The **backup** command specifies that this router is a VRRP Backup for virtual router VRID1. The IP address entered with the **ip-address** command is the same IP address as the one entered when configuring Router1. In this case, the IP address cannot also exist on Router2, but the interface on which you are configuring the VRID Backup must have an IP address in the same subnet. By entering the same IP address as the one associated with this VRID on the Owner, you are configuring the Backup to back up the address, but you are not duplicating the address.

NOTE: When you configure a Backup router, the router interface on which you are configuring the VRID must have a real IP address that is in the same subnet as the address associated with the VRID by the Owner. However, the address cannot be the same.

The **priority** parameter establishes the router's VRRP priority in relation to the other VRRP router(s) in this virtual router. The **track-priority** parameter specifies the new VRRP priority that the router receives for this VRID if the interface goes down. See "Track Ports and Track Priority" on page 37-4.

The **activate** command activates the VRID configuration on this interface. The interface does not provide backup service for the virtual IP address until you activate the VRRP configuration.

Syntax: router vrrp

Syntax: ip vrrp vrid <vrid>

Syntax: owner [track-priority <value>]

Syntax: backup [priority <value>] [track-priority <value>]

Syntax: track-port ethernet [<slotnum>/]<portnum> | ve <num>

Syntax: ip-address <ip-addr>

Syntax: activate

VRRPE Example

To implement the VRRPE configuration shown in Figure 37.3 on page 37-7, use the following CLI method.

Configuring Router1

To configure VRRP Router1 in Figure 37.3 on page 37-7, enter the following commands:

```
Router1(config)#router vrrp-extended
Router1(config)#interface ethernet 1/6
Router1(config-if-1/6)#ip address 192.53.5.2/24
Router1(config-if-1/6)#ip vrrp-extended vrid 1
Router1(config-if-1/6-vrid-1)#backup priority 110 track-priority 20
Router1(config-if-1/6-vrid-1)#track-port ethernet 2/4
Router1(config-if-1/6-vrid-1)#ip-address 192.53.5.254
Router1(config-if-1/6-vrid-1)#activate
VRRP router 1 for this interface is activating
Router1(config-if-1/6-vrid-1)#exit
Router1(config)#interface ethernet 1/6
Router1(config-if-1/6)#ip vrrp-extended vrid 2
Router1(config-if-1/6-vrid-1)#backup priority 100 track-priority 20
Router1(config-if-1/6-vrid-1)#track-port ethernet 2/4
Router1(config-if-1/6-vrid-1)#ip-address 192.53.5.253
Router1(config-if-1/6-vrid-1)#activate
VRRP router 2 for this interface is activating
```

NOTE: The address you enter with the **ip-address** command cannot be the same as a real IP address configured on the interface.

Configuring Router2

To configure Router2, enter the following commands:

```
Router1(config)#router vrrp-extended
Router1(config)#interface ethernet 5/1
Router1(config-if-5/1)#ip address 192.53.5.3/24
Router1(config-if-5/1)#ip vrrp-extended vrid 1
Router1(config-if-5/1-vrid-1)#backup priority 100 track-priority 20
Router1(config-if-5/1-vrid-1)#track-port ethernet 3/2
Router1(config-if-5/1-vrid-1)#ip-address 192.53.5.254
Router1(config-if-5/1-vrid-1)#activate
VRRP router 1 for this interface is activating
```

```
Router1(config-if-5/1-vrid-1)#exit
Router1(config)#interface ethernet 5/1
Router1(config-if-5/1)#ip vrrp-extended vrid 2
Router1(config-if-5/1-vrid-1)#backup priority 110 track-priority 20
Router1(config-if-5/1-vrid-1)#track-port ethernet 2/4
Router1(config-if-5/1-vrid-1)#ip-address 192.53.5.253
Router1(config-if-5/1-vrid-1)#activate
VRRP router 2 for this interface is activating
```

The **backup** command specifies that this router is a VRRPE Backup for virtual router VRID1. The IP address entered with the **ip-address** command is the same IP address as the one entered when configuring Router1. In this case, the IP address cannot also exist on Router2, but the interface on which you are configuring the VRID Backup must have an IP address in the same subnet. By entering the same IP address as the one associated with this VRID on the Owner, you are configuring the Backup to back up the address, but you are not duplicating the address.

NOTE: When you configure a Backup router, the router interface on which you are configuring the VRID must have a real IP address that is in the same subnet as the address associated with the VRID by the Owner. However, the address cannot be the same.

The **priority** parameter establishes the router's VRRPE priority in relation to the other VRRPE router(s) in this virtual router. The **track-priority** parameter specifies the new VRRPE priority that the router receives for this VRID if the interface goes down. See "Track Ports and Track Priority" on page 37-4.

The **activate** command activates the VRID configuration on this interface. The interface does not provide backup service for the virtual IP address until you activate the VRRPE configuration. Alternatively, you can use the **enable** command. The **activate** and **enable** commands do the same thing.

Syntax: router vrrp-extended

Syntax: ip vrrp-extended vrid <vrid>

Syntax: backup [priority <value>] [track-priority <value>]

Syntax: track-port ethernet [<slotnum>/]<portnum> | ve <num>

Syntax: ip-address <ip-addr>

Syntax: activate

Chapter 38
Configuring BGP4

This chapter provides details on how to configure ***Border Gateway Protocol version 4 (BGP4)*** on Foundry products using the CLI.

BGP4 is described in RFC 1771. The Foundry implementation fully complies with RFC 1771. The Foundry BGP4 implementation also supports the following RFCs:

- RFC 1745 (OSPF Interactions)
- RFC 1997 (BGP Communities Attributes)
- RFC 2385 (TCP MD5 Signature Option)
- RFC 2439 (Route Flap Dampening)
- RFC 2796 (Route Reflection)
- RFC 2842 (Capability Advertisement)
- RFC 3065 (BGP4 Confederations)

To display BGP4 configuration information and statistics, see “Displaying BGP4 Information” on page 38-67.

This chapter shows the commands you need in order to configure the Foundry Layer 3 Switch for BGP4.

NOTE: Your Layer 3 Switch’s management module must have 32MB or higher to run BGP4.

Overview of BGP4

Platform Support:

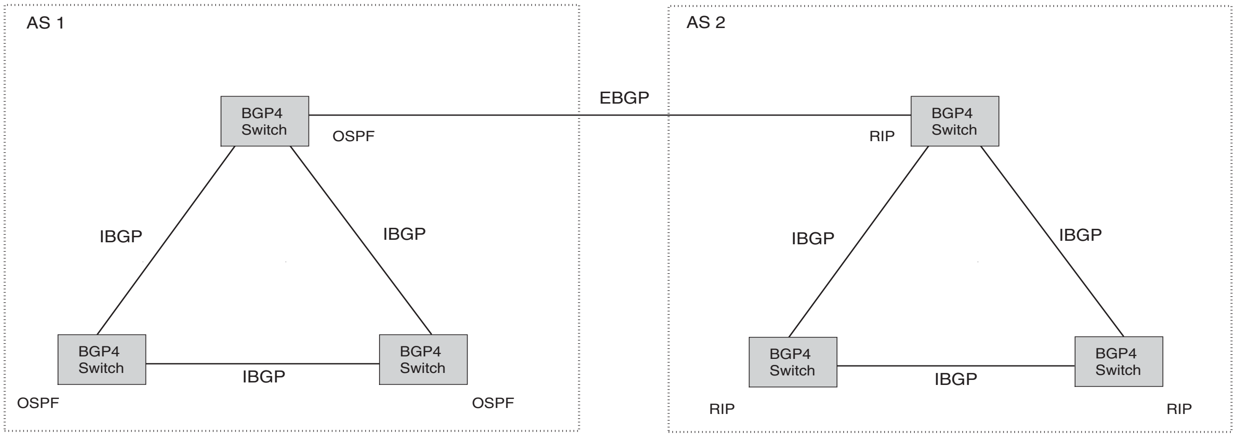
- FESX devices running software release 02.1.01 and later
- FSX devices running software release 02.2.00 and later

BGP4 is the standard Exterior Gateway Protocol (EGP) used on the Internet to route traffic between ***Autonomous Systems (AS)*** and to maintain loop-free routing. An autonomous system is a collection of networks that share the same routing and administration characteristics. For example, a corporate intranet consisting of several networks under common administrative control might be considered an AS. The networks in an AS can but do not need to run the same routing protocol to be in the same AS, nor do they need to be geographically close.

Routers within an AS can use different Interior Gateway Protocols (IGPs) such as RIP and OSPF to communicate with one another. However, for routers in different ASs to communicate, they need to use an EGP. BGP4 is the standard EGP used by Internet routers and therefore is the EGP implemented on Foundry Layer 3 Switches.

Figure 38.1 on page 38-2 shows a simple example of two BGP4 ASs. Each AS contains three BGP4 switches. All of the BGP4 switches within an AS communicate using IBGP. BGP4 switches communicate with other ASs using EBGP. Notice that each of the switches also is running an Interior Gateway Protocol (IGP). The switches in AS1 are running OSPF and the switches in AS2 are running RIP. Foundry Layer 3 Switches can be configured to redistribute routes among BGP4, RIP, and OSPF. They also can redistribute static routes.

Figure 38.1 Example BGP4 ASs



Relationship Between the BGP4 Route Table and the IP Route Table

The Foundry Layer 3 Switch's BGP4 route table can have multiple routes to the same destination, which are learned from different BGP4 neighbors. A BGP4 neighbor is another switch that also is running BGP4. BGP4 neighbors communicate using Transmission Control Protocol (TCP) port 179 for BGP communication. When you configure the Foundry Layer 3 Switch for BGP4, one of the configuration tasks you perform is to identify the Layer 3 Switch's BGP4 neighbors.

Although a Layer 3 Switch's BGP4 route table can have multiple routes to the same destination, the BGP4 protocol evaluates the routes and chooses only one of the routes to send to the IP route table. The route that BGP4 chooses and sends to the IP route table is the ***preferred route*** and will be used by the Foundry Layer 3 Switch. If the preferred route goes down, BGP4 updates the route information in the IP route table with a new BGP4 preferred route.

NOTE: If IP load sharing is enabled and you enable multiple equal-cost paths for BGP4, BGP4 can select more than one equal-cost path to a destination.

A BGP4 route consists of the following information:

- Network number (prefix) – A value comprised of the network mask bits and an IP address (<IP address>/<mask bits>); for example, 192.215.129.0/18 indicates a network mask of 18 bits applied to the IP address 192.215.129.0. When a BGP4 Layer 3 Switch advertises a route to one of its neighbors, the route is expressed in this format.
- AS-path – A list of the other ASs through which a route passes. BGP4 routers can use the AS-path to detect and eliminate routing loops. For example, if a route received by a BGP4 router contains the AS that the router is in, the router does not add the route to its own BGP4 table. (The BGP4 RFCs refer to the AS-path as "AS_PATH".)
- Additional path attributes – A list of additional parameters that describe the route. The route origin and next hop are examples of these additional path attributes.

NOTE: The Layer 3 Switch re-advertises a learned best BGP4 route to the Layer 3 Switch's neighbors even when the software does not select that route for installation in the IP route table. The best BGP4 route is the route that the software selects based on comparison of the BGP4 route path's attributes.

After a Foundry Layer 3 Switch successfully negotiates a BGP4 session with a neighbor (a BGP4 peer), the Foundry Layer 3 Switch exchanges complete BGP4 route tables with the neighbor. After this initial exchange, the Foundry Layer 3 Switch and all other RFC 1771-compliant BGP4 routers send UPDATE messages to inform neighbors of new, changed, or no longer feasible routes. BGP4 routers do not send regular updates. However, if configured to do so, a BGP4 router does regularly send KEEPALIVE messages to its peers to maintain BGP4 sessions with them if the router does not have any route information to send in an UPDATE message. See "BGP4 Message Types" on page 38-4 for information about BGP4 messages.

How BGP4 Selects a Path for a Route

When multiple paths for the same route are known to a BGP4 router, the router uses the following algorithm to weigh the paths and determine the optimal path for the route. The optimal path depends on various parameters, which can be modified. (See "Optional Configuration Tasks" on page 38-20.)

1. Is the next hop accessible though an Interior Gateway Protocol (IGP) route? If not, ignore the route.

   NOTE: The device does not use the default route to resolve BGP4 next hop. Also see "Enabling Next-Hop Recursion" on page 38-25.

2. Use the path with the largest weight.
3. If the weights are the same, prefer the route with the largest local preference.
4. If the routes have the same local preference, prefer the route that was originated locally (by this BGP4 Layer 3 Switch).
5. If the local preferences are the same, prefer the route with the shortest AS-path. An AS-SET counts as 1. A confederation path length, if present, is not counted as part of the path length.
6. If the AS-path lengths are the same, prefer the route with the lowest origin type. From low to high, route origin types are valued as follows:
   - IGP is lowest
   - EGP is higher than IGP but lower than INCOMPLETE
   - INCOMPLETE is highest
7. If the routes have the same origin type, prefer the route with the lowest MED. For a definition of MED, see "Configuring the Layer 3 Switch To Always Compare Multi-Exit Discriminators (MEDs)" on page 38-30.

   0BGP4 compares the MEDs of two otherwise equivalent paths ***if and only if*** the routes were learned from the same neighboring AS. This behavior is called ***deterministic MED***. 0Deterministic MED is always enabled and cannot be disabled.

   In addition, you can enable the Layer 3 Switch to always compare the MEDs, regardless of the AS information in the paths. To enable this comparison, enter the **always-compare-med** command at the BGP4 configuration level of the CLI. This option is disabled by default.

   NOTE: By default, value 0 (most favorable) is used in MED comparison when the MED attribute is not present. The default MED comparison results in the Layer 3 Switch favoring the route paths that are missing their MEDs. 0You can use the **med-missing-as-worst** command to make the Layer 3 Switch regard a BGP route with a missing MED attribute as the least favorable route, when comparing the MEDs of the routes.

   NOTE: MED comparison is not performed for internal routes originated within the local AS or confederation.

8. Prefer routes in the following order:
   - Routes received through EBGP from a BGP4 neighbor outside of the confederation
   - Routes received through EBGP from a BGP4 router within the confederation
   - Routes received through IBGP
9. If all the comparisons above are equal, prefer the route with the lowest IGP metric to the BGP4 next hop. This is the closest internal path inside the AS to reach the destination.
10. If the internal paths also are the same and BGP4 load sharing is enabled, load share among the paths. Otherwise, prefer the route that comes from the BGP4 router with the lowest router ID.

NOTE: Foundry Layer 3 Switches support BGP4 load sharing among multiple equal-cost paths. BGP4 load sharing enables the Layer 3 Switch to balance the traffic across the multiple paths instead of choosing just one path based on router ID. For EBGP routes, load sharing applies only when the paths are from neighbors within the same remote AS. EBGP paths from neighbors in different ASs are not compared.

BGP4 Message Types

BGP4 routers communicate with their neighbors (other BGP4 routers) using the following types of messages:

- OPEN
- UPDATE
- KEEPALIVE
- NOTIFICATION

OPEN Message

After a BGP4 router establishes a TCP connection with a neighboring BGP4 router, the routers exchange OPEN messages. An OPEN message indicates the following:

- BGP version – Indicates the version of the protocol that is in use on the router. BGP version 4 supports Classless Interdomain Routing (CIDR) and is the version most widely used in the Internet. Version 4 also is the only version supported on Foundry Layer 3 Switches.
- AS number – A two-byte number that identifies the AS to which the BGP4 router belongs.
- Hold Time – The number of seconds a BGP4 router will wait for an UPDATE or KEEPALIVE message (described below) from a BGP4 neighbor before assuming that the neighbor is dead. BGP4 routers exchange UPDATE and KEEPALIVE messages to update route information and maintain communication. If BGP4 neighbors are using different Hold Times, the lowest Hold Time is used by the neighbors. If the Hold Time expires, the BGP4 router closes its TCP connection to the neighbor and clears any information it has learned from the neighbor and cached.

  You can configure the Hold Time to be 0, in which case a BGP4 router will consider its neighbors to always be up. For directly-attached neighbors, you can configure the Foundry Layer 3 Switch to immediately close the TCP connection to the neighbor and clear entries learned from an EBGP neighbor if the interface to that neighbor goes down. This capability is provided by the fast external fallover feature, which is disabled by default.
- BGP Identifier – The router ID. The BGP Identifier (router ID) identifies the BGP4 router to other BGP4 routers. Foundry Layer 3 Switches use the same router ID for OSPF and BGP4. If you do not set a router ID, the software uses the IP address on the lowest numbered loopback interface configured on the router. If the Layer 3 Switch does not have a loopback interface, the default router ID is the lowest numbered IP address configured on the device. For more information or to change the router ID, see “Changing the Router ID” on page 29-24.
- Parameter list – An optional list of additional parameters used in peer negotiation with BGP4 neighbors.

UPDATE Message

After BGP4 neighbors establish a BGP4 connection over TCP and exchange their BGP4 routing tables, they do not send periodic routing updates. Instead, a BGP4 neighbor sends an update to its neighbor when it has a new route to advertise or routes have changed or become unfeasible. An UPDATE message can contain the following information:

- Network Layer Reachability Information (NLRI) – The mechanism by which BGP4 supports Classless Interdomain Routing (CIDR). An NLRI entry consists of an IP prefix that indicates a network being advertised by the UPDATE message. The prefix consists of an IP network number and the length of the network portion of the number. For example, an UPDATE message with the NLRI entry 192.215.129.0/18 indicates a route to IP network 192.215.129.0 with network mask 255.255.192.0. The binary equivalent of this mask is 18 consecutive one bits, thus “18” in the NLRI entry.
- Path attributes – Parameters that indicate route-specific information such as path information, route preference, next hop values, and aggregation information. BGP4 uses the path attributes to make filtering and routing decisions.
- Unreachable routes – A list of routes that have been in the sending router’s BGP4 table but are no longer feasible. The UPDATE message lists unreachable routes in the same format as new routes: <IP address>/<CIDR prefix>.

KEEPALIVE Message

BGP4 routers do not regularly exchange UPDATE messages to maintain the BGP4 sessions. For example, if a Layer 3 Switch configured to perform BGP4 routing has already sent the latest route information to its peers in UPDATE messages, the router does not send more UPDATE messages. Instead, BGP4 routers send KEEPALIVE messages to maintain the BGP4 sessions. KEEPALIVE messages are 19 bytes long and consist only of a message header; they contain no routing data.

BGP4 routers send KEEPALIVE messages at a regular interval, the Keep Alive Time. The default Keep Alive Time on Foundry Layer 3 Switches is 60 seconds.

A parameter related to the Keep Alive Time is the Hold Time. A BGP4 router’s Hold Time determines how many seconds the router will wait for a KEEPALIVE or UPDATE message from a BGP4 neighbor before deciding that the neighbor is dead. The Hold Time is negotiated when BGP4 routers exchange OPEN messages; the lower Hold Time is then used by both neighbors. For example, if BGP4 Router A sends a Hold Time of 5 seconds and BGP4 Router B sends a Hold Time of 4 seconds, both routers use 4 seconds as the Hold Time for their BGP4 session. The default Hold Time is 180 seconds. Generally, the Hold Time is configured to three times the value of the Keep Alive Time.

If the Hold Time is 0, a BGP4 router assumes that its neighbor is alive regardless of how many seconds pass between receipt of UPDATE or KEEPALIVE messages.

NOTIFICATION Message

When you close the router’s BGP4 session with a neighbor, or the router detects an error in a message received from the neighbor, or an error occurs on the router, the router sends a NOTIFICATION message to the neighbor. No further communication takes place between the BGP4 router that sent the NOTIFICATION and the neighbor(s) that received the NOTIFICATION.

Basic Configuration and Activation for BGP4

BGP4 is disabled by default. To enable BGP4 and place your Foundry Layer 3 Switch into service as a BGP4 router, you must perform at least the following steps:

1. Enable the BGP4 protocol.
2. Set the local AS number.

NOTE: You must specify the local AS number for BGP4 to become functional.

3. Add each BGP4 neighbor (peer BGP4 router) and identify the AS the neighbor is in.

4. Save the BGP4 configuration information to the system configuration file.

NOTE: By default, the Foundry router ID is the IP address configured on the lowest numbered loopback interface. If the Layer 3 Switch does not have a loopback interface, the default router ID is the lowest numbered IP interface address configured on the device. For more information or to change the router ID, see "Changing the Router ID" on page 29-24. If you change the router ID, all current BGP4 sessions are cleared.

```
FastIron> enable
FastIron#configure terminal
FastIron(config)#router bgp
BGP4: Please configure 'local-as' parameter in order to enable BGP4.
FastIron(config-bgp-router)#local-as 10
FastIron(config-bgp-router)#neighbor 209.157.23.99 remote-as 100
FastIron(config-bgp-router)#write memory
```

NOTE: When BGP4 is enabled on a Foundry Layer 3 Switch, you do not need to reset the system. The protocol is activated as soon as you enable it. Moreover, the router begins a BGP4 session with a BGP4 neighbor as soon as you add the neighbor.

Note Regarding Disabling BGP4

If you disable BGP4, the Layer 3 Switch removes all the running configuration information for the disabled protocol from the running-config. To restore the BGP4 configuration, you must reload the software to load the configuration from the startup-config. Moreover, when you save the configuration to the startup-config file after disabling the protocol, all the configuration information for the disabled protocol is removed from the startup-config file.

The CLI displays a warning message such as the following:

```
FastIron(config-bgp-router)#no router bgp
router bgp mode now disabled. All bgp config data will be lost when writing to
flash!
```

If you are testing a BGP4 configuration and are likely to disable and re-enable the protocol, you might want to make a backup copy of the startup-config file containing the protocol's configuration information. This way, if you remove the configuration information by saving the configuration after disabling the protocol, you can restore the configuration by copying the backup copy of the startup-config file onto the flash memory.

NOTE: To disable BGP4 without losing the BGP4 configuration information, remove the local AS (for example, by entering the **no local-as** <num> command). In this case, BGP4 retains the other configuration information but is not operational until you set the local AS again.

BGP4 Parameters

You can modify or set the following BGP4 parameters.

- Optional – Define the router ID. (The same router ID also is used by OSPF.)
- Required – Specify the local AS number.
- Optional – Add a loopback interface for use with neighbors.
- Required – Identify BGP4 neighbors.
- Optional – Change the Keep Alive Time and Hold Time.
- Optional – Change the update timer for route changes.
- Optional – Enable fast external fallover.
- Optional – Specify a list of individual networks in the local AS to be advertised to remote ASs using BGP4.

- Optional – Change the default local preference for routes.
- Optional – Enable the default route (default-information-originate).
- Optional – Enable use of a default route to resolve a BGP4 next-hop route.
- Optional – Change the default MED (metric).
- Optional – Enable next-hop recursion.
- Optional – Change the default administrative distances for EBGP, IBGP, and locally originated routes.
- Optional – Require the first AS in an Update from an EBGP neighbor to be the neighbor's AS.
- Optional – Change MED comparison parameters.
- Optional – Disable comparison of the AS-Path length.
- Optional – Enable comparison of the router ID.
- Optional – Enable auto summary to summarize routes at an IP class boundary (A, B, or C).
- Optional – Aggregate routes in the BGP4 route table into CIDR blocks.
- Optional – Configure the router as a BGP4 router reflector.
- Optional – Configure the Layer 3 Switch as a member of a BGP4 confederation.
- Optional – Change the default metric for routes that BGP4 redistributes into RIP or OSPF.
- Optional – Change the parameters for RIP, OSPF, or static routes redistributed into BGP4.
- Optional – Change the number of paths for BGP4 load sharing.
- Optional – Change other load-sharing parameters
- Optional – Define BGP4 address filters.
- Optional – Define BGP4 AS-path filters.
- Optional – Define BGP4 community filters.
- Optional – Define IP prefix lists.
- Optional – Define neighbor distribute lists.
- Optional – Define BGP4 route maps for filtering routes redistributed into RIP and OSPF.
- Optional – Define route flap dampening parameters.

NOTE: When using CLI, you set global level parameters at the BGP CONFIG Level of the CLI. You can reach the BGP CONFIG level by entering **router bgp…** at the global CONFIG level.

When Parameter Changes Take Effect

Some parameter changes take effect immediately while others do not take full effect until the router's sessions with its neighbors are reset. Some parameters do not take effect until the router is rebooted.

Immediately

The following parameter changes take effect immediately:

- Enable or disable BGP.
- Set or change the local AS.
- Add neighbors.
- Change the update timer for route changes.
- Disable or enable fast external fallover.

- Specify individual networks that can be advertised.
- Change the default local preference, default information originate setting, or administrative distance.
- Enable or disable use of a default route to resolve a BGP4 next-hop route.
- Enable or disable MED (metric) comparison.
- Require the first AS in an Update from an EBGP neighbor to be the neighbor's AS.
- Change MED comparison parameters.
- Disable comparison of the AS-Path length.
- Enable comparison of the router ID.
- Enable next-hop recursion.
- Enable or disable auto summary.
- Change the default metric.
- Disable or re-enable route reflection.
- Configure confederation parameters.
- Disable or re-enable load sharing.
- Change the maximum number of load-sharing paths.
- Change other load-sharing parameters.
- Define route flap dampening parameters.
- Add, change, or negate redistribution parameters (except changing the default MED; see below).
- Add, change, or negate route maps (when used by the **network** command or a redistribution command).

After Resetting Neighbor Sessions

The following parameter changes take effect only after the router's BGP4 sessions are cleared, or reset using the "soft" clear option. (See "Closing or Resetting a Neighbor Session" on page 38-107.)

- Change the Hold Time or Keep Alive Time.
- Aggregate routes.
- Add, change, or negate filter tables.

After Disabling and Re-Enabling Redistribution

The following parameter change takes effect only after you disable and then re-enable redistribution:

- Change the default MED (metric).

Memory Considerations

BGP4 handles a very large number of routes and therefore requires a lot of memory. For example, in a typical configuration with just a single BGP4 neighbor, a BGP4 router may need to be able to hold up to 80,000 routes. Many configurations, especially those involving more than one neighbor, can require the router to hold even more routes. Foundry Layer 3 Switches provide dynamic memory allocation for BGP4 data. These devices automatically allocate memory when needed to support BGP4 neighbors, routes, and route attribute entries. Dynamic memory allocation is performed automatically by the software and does not require a reload.

Table 38.1 lists the maximum total amount of system memory (DRAM) BGP4 can use. The maximum depends on the total amount of system memory on the device.

Table 38.1: Maximum Memory Usage

Platform	Maximum Memory BGP4 Can Use
FESX with 128 MB	30 MB
FSX with Management 1 module with 256 MB	130 MB
FSX with Management 2 module with 512 MB	400 MB

The memory amounts listed in the table are for all BGP4 data, including routes received from neighbors, BGP route advertisements (routes sent to neighbors), and BGP route attribute entries. The routes sent to and received from neighbors use the most BGP4 memory. Generally, the actual limit to the number of neighbors, routes, or route attribute entries the device can accommodate depends on how many routes the Layer 3 Switch sends to and receives from the neighbors.

In some cases, where most of the neighbors do not send or receive a full BGP route table (about 80,000 routes), the memory can support a larger number of BGP4 neighbors. However, if most of the BGP4 neighbors send or receive full BGP route tables, the number of BGP neighbors the memory can support is less than in configurations where the neighbors send smaller route tables.

As a guideline, Layer 3 Switches with a 512 MB Management 4 module can accommodate 150 – 200 neighbors, with the assumption that the Layer 3 Switch receives about one million routes total from all neighbors and sends about eight million routes total to neighbors. For each additional one million incoming routes, the capacity for outgoing routes decreases by around two million.

Memory Configuration Options Obsoleted by Dynamic Memory

Devices that support dynamic BGP4 memory allocation do not require or even support static configuration of memory for BGP4 neighbors, routes, or route attributes. Consequently, the following CLI commands and equivalent Web management options are not supported on these devices:

- **max-neighbors** <num>
- **max-routes** <num>
- **max-attribute-entries** <num>

If you boot a device that has a startup-config file that contains these commands, the software ignores the commands and uses dynamic memory allocation for BGP4. The first time you save the device's running configuration (running-config) to the startup-config file, the commands are removed from the file.

Basic Configuration Tasks

The following sections describe how to perform the configuration tasks that are required to use BGP4 on the Foundry Layer 3 Switch. You can modify many parameters in addition to the ones described in this section. See "Optional Configuration Tasks" on page 38-20.

Enabling BGP4 on the Router

When you enable BGP4 on the router, BGP4 is automatically activated. To enable BGP4 on the router, enter the following commands:

```
FastIron> enable
FastIron#configure terminal
FastIron(config)#router bgp
```

```
BGP4: Please configure 'local-as' parameter in order to enable BGP4.
FastIron(config-bgp-router)#local-as 10
FastIron(config-bgp-router)#neighbor 209.157.23.99 remote-as 100
FastIron(config-bgp-router)#write memory
```

Changing the Router ID

The OSPF and BGP4 protocols use router IDs to identify the routers that are running the protocols. A router ID is a valid, unique IP address and sometimes is an IP address configured on the router. The router ID cannot be an IP address in use by another device.

By default, the router ID on a Foundry Layer 3 Switch is one of the following:

- If the router has loopback interfaces, the default router ID is the IP address configured on the lowest numbered loopback interface configured on the Layer 3 Switch. For example, if you configure loopback interfaces 1, 2, and 3 as follows, the default router ID is 9.9.9.9/24:
  - Loopback interface 1, 9.9.9.9/24
  - Loopback interface 2, 4.4.4.4/24
  - Loopback interface 3, 1.1.1.1/24
- If the device does not have any loopback interfaces, the default router ID is the lowest numbered IP interface address configured on the device.

NOTE: Foundry Layer 3 Switches use the same router ID for both OSPF and BGP4. If the router is already configured for OSPF, you may want to use the router ID that is already in use on the router rather than set a new one. To display the router ID, enter the **show ip** CLI command at any CLI level.

To change the router ID, enter a command such as the following:

```
FastIron(config)#ip router-id 209.157.22.26
```

Syntax: ip router-id <ip-addr>

The <ip-addr> can be any valid, unique IP address.

NOTE: You can specify an IP address used for an interface on the Foundry Layer 3 Switch, but do not specify an IP address in use by another device.

Setting the Local AS Number

The local AS number identifies the AS the Foundry BGP4 router is in. The AS number can be from 1 – 65535. There is no default. AS numbers 64512 – 65535 are the well-known private BGP4 AS numbers and are not advertised to the Internet community.

To set the local AS number, enter commands such as the following:

```
FastIron(config)#router bgp
BGP4: Please configure 'local-as' parameter in order to enable BGP4.
FastIron(config-bgp-router)#local-as 10
FastIron(config-bgp-router)#write memory
```

Syntax: [no] local-as <num>

The <num> parameter specifies the local AS number.

Adding a Loopback Interface

You can configure the router to use a loopback interface instead of a specific port or virtual routing interface to communicate with a BGP4 neighbor. A loopback interface adds stability to the network by working around route flap problems that can occur due to unstable links between the router and its neighbors.

Loopback interfaces are always up, regardless of the states of physical interfaces. Loopback interfaces are especially useful for IBGP neighbors (neighbors in the same AS) that are multiple hops away from the router. When you configure a BGP4 neighbor on the router, you can specify whether the router uses the loopback interface to communicate with the neighbor. As long as a path exists between the router and its neighbor, BGP4 information can be exchanged. The BGP4 session is not associated with a specific link but instead is associated with the virtual interfaces.

You can add up to 24 IP addresses to each loopback interface.

NOTE: If you configure the Foundry Layer 3 Switch to use a loopback interface to communicate with a BGP4 neighbor, the peer IP address on the remote router pointing to your loopback address must be configured.

To add a loopback interface, enter commands such as those shown in the following example:

```
FastIron(config-bgp-router)#exit
FastIron(config)#int loopback 1
FastIron(config-lbif-1)#ip address 10.0.0.1/24
```

Syntax: interface loopback <num>

The <num> value can be from 1 – 8 on Chassis Layer 3 Switches. The value can be from 1 – 4 on the Compact Layer 3 Switch.

Adding BGP4 Neighbors

The BGP4 protocol does not contain a peer discovery process. Therefore, for each of the router’s BGP4 neighbors (peers), you must indicate the neighbor’s IP address and the AS each neighbor is in. Neighbors that are in different ASs communicate using EBGP. Neighbors within the same AS communicate using IBGP.

NOTE: If the Layer 3 Switch has multiple neighbors with similar attributes, you can simplify configuration by configuring a peer group, then adding individual neighbors to it. The configuration steps are similar, except you specify a peer group name instead of a neighbor IP address when configuring the neighbor parameters, then add individual neighbors to the peer group. See “Adding a BGP4 Peer Group” on page 38-16.

NOTE: The Layer 3 Switch attempts to establish a BGP4 session with a neighbor as soon as you enter a command specifying the neighbor’s IP address. If you want to completely configure the neighbor parameters before the Layer 3 Switch establishes a session with the neighbor, you can administratively shut down the neighbor. See “Administratively Shutting Down a Session with a BGP4 Neighbor” on page 38-19.

To add a BGP4 neighbor with IP address 209.157.22.26, enter the following command:

```
FastIron(config-bgp-router)#neighbor 209.157.22.26
```

The neighbor’s <ip-addr> must be a valid IP address.

The **neighbor** command has some additional parameters, as shown in the following syntax:

Syntax: [no] neighbor <ip-addr> | <peer-group-name>
[advertisement-interval <num>]
[capability orf prefixlist [send | receive]]
[default-originate [route-map <map-name>]]
[description <string>]
[distribute-list in | out <num,num,...> | <acl-num> in | out]
[ebgp-multihop [<num>]]
[filter-list in | out <num,num,...> | <acl-num> in | out | weight]
[maximum-prefix <num> [<threshold>] [teardown]]
[next-hop-self]
[nlri multicast | unicast | multicast unicast]
[password [0 | 1] <string>]
[prefix-list <string> in | out]
[remote-as <as-number>]
[remove-private-as]
[route-map in | out <map-name>]
[route-reflector-client]
[send-community]
[soft-reconfiguration inbound]
[shutdown]
[timers keep-alive <num> hold-time <num>]
[unsuppress-map <map-name>]
[update-source <ip-addr> | ethernet [<slotnum>/]<portnum> | loopback <num> | ve <num>]
[weight <num>]

The <ip-addr> | <peer-group-name> parameter indicates whether you are configuring an individual neighbor or a peer group. If you specify a neighbor's IP address, you are configuring that individual neighbor. If you specify a peer group name, you are configuring a peer group. See "Adding a BGP4 Peer Group" on page 38-16.

advertisement-interval <num> specifies the minimum delay (in seconds) between messages to the specified neighbor. The default is 30 for EBGP neighbors (neighbors in other ASs). The default is 5 for IBGP neighbors (neighbors in the same AS). The range is 0 – 600.

NOTE: The Layer 3 Switch applies the advertisement interval only under certain conditions. The Layer 3 Switch does not apply the advertisement interval when sending initial updates to a BGP4 neighbor. As a result, the Layer 3 Switch sends the updates one immediately after another, without waiting for the advertisement interval.

capability orf prefixlist [**send** | **receive**] configures cooperative router filtering. The **send** | **receive** parameter specifies the support you are enabling:

- **send** – The Layer 3 Switch sends the IP prefix lists as Outbound Route Filters (ORFs) to the neighbor.
- **receive** – The Layer 3 Switch accepts filters as Outbound Route Filters (ORFs) from the neighbor.

If you do not specify the capability, both capabilities are enabled. The **prefixlist** parameter specifies the type of filter you want to send to the neighbor.

For more information, see "Configuring Cooperative BGP4 Route Filtering" on page 38-58.

NOTE: The current release supports cooperative filtering only for filters configured using IP prefix lists.

default-originate [**route-map** <map-name>] configures the Layer 3 Switch to send the default route 0.0.0.0 to the neighbor. If you use the route-map <map-name> parameter, the route map injects the default route conditionally, based on the match conditions in the route map.

description <string> specifies a name for the neighbor. You can enter an alphanumeric text string up to 80 characters long.

distribute-list in | out <num,num,...> specifies a distribute list to be applied to updates to or from the specified neighbor. The **in | out** keyword specifies whether the list is applied on updates received from the neighbor or sent to the neighbor. The <num,num,...> parameter specifies the list of address-list filters. The router applies the filters in the order in which you list them and stops applying the filters in the distribute list when a match is found.

Alternatively, you can specify **distribute-list** <acl-num> **in | out** to use an IP ACL instead of a distribute list. In this case, <acl-num> is an IP ACL.

NOTE: By default, if a route does not match any of the filters, the Layer 3 Switch denies the route. To change the default behavior, configure the last filter as “permit any any”.

NOTE: The address filter must already be configured. See “Filtering Specific IP Addresses” on page 38-43.

ebgp-multihop [<num>] specifies that the neighbor is more than one hop away and that the session type with the neighbor is thus EBGP-multihop. This option is disabled by default. The <num> parameter specifies the TTL you are adding for the neighbor. You can specify a number from 0 – 255. The default is 0. If you leave the EBGP TTL value set to 0, the software uses the IP TTL value.

filter-list in | out <num,num,...> specifies an AS-path filter list or a list of AS-path ACLs. The **in | out** keyword specifies whether the list is applied on updates received from the neighbor or sent to the neighbor. If you specify **in** or **out**, The <num,num,...> parameter specifies the list of AS-path filters. The router applies the filters in the order in which you list them and stops applying the filters in the AS-path filter list when a match is found. The **weight** <num> parameter specifies a weight that the Layer 3 Switch applies to routes received from the neighbor that match the AS-path filter or ACL. You can specify a number from 0 – 65535.

Alternatively, you can specify **filter-list** <acl-num> **in | out | weight** to use an AS-path ACL instead of an AS-path filter list. In this case, <acl-num> is an AS-path ACL.

NOTE: By default, if an AS-path does not match any of the filters or ACLs, the Layer 3 Switch denies the route. To change the default behavior, configure the last filter or ACL as “permit any any”.

NOTE: The AS-path filter or ACL must already be configured. See “Filtering AS-Paths” on page 38-44.

maximum-prefix <num> specifies the maximum number of IP network prefixes (routes) that can be learned from the specified neighbor or peer group. You can specify a value from 0 – 4294967295. The default is 0 (unlimited).

- The <num> parameter specifies the maximum number. You can specify a value from 0 – 4294967295. The default is 0 (unlimited).
- The <threshold> parameter specifies the percentage of the value you specified for the **maximum-prefix** <num>, at which you want the software to generate a Syslog message. You can specify a value from 1 (one percent) to 100 (100 percent). The default is 100.
- The **teardown** parameter tears down the neighbor session if the maximum-prefix limit is exceeded. The session remains shutdown until you clear the prefixes using the **clear ip bgp neighbor all** or **clear ip bgp neighbor** <ip-addr> command, or change the neighbor’s maximum-prefix configuration. The software also generates a Syslog message.

next-hop-self specifies that the router should list itself as the next hop in updates sent to the specified neighbor. This option is disabled by default.

The **nlri multicast | unicast | multicast unicast** parameter specifies whether the neighbor is a multicast neighbor or a unicast neighbor. For MBGP, you must specify **multicast**. Optionally, you also can specify **unicast** if you want the Layer 3 Switch to exchange unicast (BGP4) routes as well as multicast routes with the neighbor. The default is **unicast** only.

password [**0 | 1**] <string> specifies an MD5 password for securing sessions between the Layer 3 Switch and the neighbor. You can enter a string up to 80 characters long. The string can contain any alphanumeric characters, but the first character cannot be a number. If the password contains a number, do not enter a space following the number.

The **0 | 1** parameter is the encryption option, which you can omit (the default) or which can be one of the following.

- **0** – Disables encryption for the authentication string you specify with the command. The password or string is shown as clear text in the output of commands that display neighbor or peer group configuration information.
- **1** – Assumes that the authentication string you enter is the encrypted form, and decrypts the value before using it.

For more information, see “Encryption of BGP4 MD5 Authentication Keys” on page 38-15.

NOTE: If you want the software to assume that the value you enter is the clear-text form, and to encrypt display of that form, do not enter 0 or 1. Instead, omit the encryption option and allow the software to use the default behavior.

If you specify encryption option 1, the software assumes that you are entering the encrypted form of the password or authentication string. In this case, the software decrypts the password or string you enter before using the value for authentication. If you accidentally enter option 1 followed by the clear-text version of the password or string, authentication will fail because the value used by the software will not match the value you intended to use.

prefix-list <string> **in | out** specifies an IP prefix list. You can use IP prefix lists to control routes to and from the neighbor. IP prefix lists are an alternative method to AS-path filters. The **in | out** keyword specifies whether the list is applied on updates received from the neighbor or sent to the neighbor. You can configure up to 1000 prefix list filters. The filters can use the same prefix list or different prefix lists. To configure an IP prefix list, see “Defining IP Prefix Lists” on page 38-49.

remote-as <as-number> specifies the AS the remote neighbor is in. The <as-number> can be a number from 1 – 65535. There is no default.

remove-private-as configures the router to remove private AS numbers from UPDATE messages the router sends to this neighbor. The router will remove AS numbers 64512 – 65535 (the well-known BGP4 private AS numbers) from the AS-path attribute in UPDATE messages the Layer 3 Switch sends to the neighbor. This option is disabled by default.

route-map in | out <map-name> specifies a route map the Layer 3 Switch will apply to updates sent to or received from the specified neighbor. The **in | out** keyword specifies whether the list is applied on updates received from the neighbor or sent to the neighbor.

NOTE: The route map must already be configured. See “Defining Route Maps” on page 38-50.

route-reflector-client specifies that this neighbor is a route-reflector client of the router. Use the parameter only if this router is going to be a route reflector. For information, see “Configuring Route Reflection Parameters” on page 38-31. This option is disabled by default.

send-community enables sending the community attribute in updates to the specified neighbor. By default, the router does not send the community attribute.

shutdown administratively shuts down the session with this neighbor. Shutting down the session allows you to completely configure the neighbor and save the configuration without actually establishing a session with the neighbor. This option is disabled by default.

soft-reconfiguration inbound enables the soft reconfiguration feature, which stores all the route updates received from the neighbor. If you request a soft reset of inbound routes, the software performs the reset by comparing the policies against the stored route updates, instead of requesting the neighbor’s BGP4 route table or resetting the session with the neighbor. See “Using Soft Reconfiguration” on page 38-102.

timers keep-alive <num> **hold-time** <num> overrides the global settings for the Keep Alive Time and Hold Time. For the Keep Alive Time, you can specify from 0 – 65535 seconds. For the Hold Time, you can specify 0 or 3 – 65535 (1 and 2 are not allowed). If you set the Hold Time to 0, the router waits indefinitely for messages from a neighbor without concluding that the neighbor is dead. The defaults for these parameters are the currently configured global Keep Alive Time and Hold Time. For more information about these parameters, see “Changing the Keep Alive Time and Hold Time” on page 38-20.

unsuppress-map <map-name> removes route dampening from a neighbor’s routes when those routes have been dampened due to aggregation. See “Removing Route Dampening from a Neighbor’s Routes Suppressed Due to Aggregation” on page 38-63.

update-source <ip-addr> | **ethernet** [<slotnum>/]<portnum> | **loopback** <num> | **ve** <num> configures the router to communicate with the neighbor through the specified interface. There is no default.

weight <num> specifies a weight the Layer 3 Switch will add to routes received from the specified neighbor. BGP4 prefers larger weights over smaller weights. The default weight is 0.

Encryption of BGP4 MD5 Authentication Keys

When you configure a BGP4 neighbor or neighbor peer group, you can specify an MD5 authentication string for authenticating packets exchanged with the neighbor or peer group of neighbors.

For added security, the software encrypts display of the authentication string by default. The software also provides an optional parameter to disable encryption of the authentication string, on an individual neighbor or peer group basis. By default, the MD5 authentication strings are displayed in encrypted format in the output of the following commands:

- **show running-config** (or **write terminal**)
- **show configuration**
- **show ip bgp config**

When encryption of the authentication string is enabled, the string is encrypted in the CLI regardless of the access level you are using.

If you display the running-config after reloading, the BGP4 commands that specify an authentication string show the string in encrypted form.

In addition, when you save the configuration to the startup-config file, the file contains the new BGP4 command syntax and encrypted passwords or strings.

NOTE: Foundry recommends that you save a copy of the startup-config file for each switch you plan to upgrade.

Encryption Example

The following commands configure a BGP4 neighbor and a peer group, and specify MD5 authentication strings (passwords) for authenticating packets exchanged with the neighbor or peer group.

```
FastIron(config-bgp-router)#local-as 2
FastIron(config-bgp-router)#neighbor xyz peer-group
FastIron(config-bgp-router)#neighbor xyz password abc
FastIron(config-bgp-router)#neighbor 10.10.200.102 peer-group xyz
FastIron(config-bgp-router)#neighbor 10.10.200.102 password test
```

Here is how the commands appear when you display the BGP4 configuration commands:

```
FastIron#show ip bgp config
Current BGP configuration:
router bgp
 local-as 2
 neighbor xyz peer-group
 neighbor xyz password 1 $!2d
 neighbor 10.10.200.102 peer-group xyz
 neighbor 10.10.200.102 remote-as 1
 neighbor 10.10.200.102 password 1 $on-o
```

Notice that the software has converted the commands that specify an authentication string into the new syntax (described below), and has encrypted display of the authentication strings.

Command Syntax

Since the default behavior does not affect the BGP4 configuration itself but does encrypt display of the authentication string, the CLI does not list the encryption options.

Syntax: [no] neighbor <ip-addr> | <peer-group-name> password [0 | 1] <string>

The <ip-addr> | <peer-group-name> parameter indicates whether you are configuring an individual neighbor or a peer group. If you specify a neighbor's IP address, you are configuring that individual neighbor. If you specify a peer group name, you are configuring a peer group.

The **password** <string> parameter specifies an MD5 authentication string for securing sessions between the Layer 3 Switch and the neighbor. You can enter a string up to 80 characters long. The string can contain any alphanumeric characters, but the first character cannot be a number. If the password contains a number, do not enter a space following the number.

The **0 | 1** parameter is the encryption option, which you can omit (the default) or which can be one of the following.

- **0** – Disables encryption for the authentication string you specify with the command. The password or string is shown as clear text in the output of commands that display neighbor or peer group configuration information.
- **1** – Assumes that the authentication string you enter is the encrypted form, and decrypts the value before using it.

NOTE: If you want the software to assume that the value you enter is the clear-text form, and to encrypt display of that form, do not enter 0 or 1. Instead, omit the encryption option and allow the software to use the default behavior.

If you specify encryption option 1, the software assumes that you are entering the encrypted form of the password or authentication string. In this case, the software decrypts the password or string you enter before using the value for authentication. If you accidentally enter option 1 followed by the clear-text version of the password or string, authentication will fail because the value used by the software will not match the value you intended to use.

Displaying the Authentication String

If you want to display the authentication string, enter the following commands:

```
FastIron(config)#enable password-display
FastIron#show ip bgp neighbors
```

The **enable password-display** command enables display of the authentication string, but only in the output of the **show ip bgp neighbors** command. Display of the string is still encrypted in the startup-config file and running-config. Enter the command at the global CONFIG level of the CLI.

NOTE: The command also displays SNMP community strings in clear text, in the output of the **show snmp server** command.

Adding a BGP4 Peer Group

A ***peer group*** is a set of BGP4 neighbors that share common parameters. Peer groups provide the following benefits:

- Simplified neighbor configuration – You can configure a set of neighbor parameters and then apply them to multiple neighbors. You do not need to individually configure the common parameters individually on each neighbor.
- Flash memory conservation – Using peer groups instead of individually configuring all the parameters for each neighbor requires fewer configuration commands in the startup-config file.

You can perform the following tasks on a peer-group basis.

- Reset neighbor sessions
- Perform soft-outbound resets (the Layer 3 Switch updates outgoing route information to neighbors but does not entirely reset the sessions with those neighbors)
- Clear BGP message statistics
- Clear error buffers

Peer Group Parameters

You can set all neighbor parameters in a peer group. When you add a neighbor to the peer group, the neighbor receives all the parameter settings you set in the group, except parameter values you have explicitly configured for the neighbor. If you do not set a neighbor parameter in the peer group and the parameter also is not set for the individual neighbor, the neighbor uses the default value.

Configuration Rules

The following rules apply to peer group configuration:

- You must configure a peer group before you can add neighbors to the peer group.
- If you remove a parameter from a peer group, the value for that parameter is reset to the default for all the neighbors within the peer group, unless you have explicitly set that parameter on individual neighbors. In this case, the value you set on the individual neighbors applies to those neighbors, while the default value applies to neighbors for which you have not explicitly set the value.

NOTE: If you enter a command to remove the remote AS parameter from a peer group, the software checks to ensure that the peer group does not contain any neighbors. If the peer group does contain neighbors, the software does not allow you to remove the remote AS. The software prevents removing the remote AS in this case so that the neighbors in the peer group that are using the remote AS do not lose connectivity to the Layer 3 Switch.

- Once you add a neighbor to a peer group, you cannot configure the following outbound parameters (the parameters governing outbound traffic) for the neighbor.
  - Default-information-originate
  - Next-hop-self
  - Outbound route map
  - Outbound filter list
  - Outbound distribute list
  - Outbound prefix list
  - Remote AS, if configured for the peer group
  - Remove private AS
  - Route reflector client
  - Send community
  - Timers
  - Update source

  If you want to change an outbound parameter for an individual neighbor, you must first remove the neighbor from the peer group. In this case, you cannot re-add the neighbor to the same peer group, but you can add the neighbor to a different peer group. All the neighbors within a peer group must have the same values for the outbound parameters. To change an outbound parameter to the same value for all neighbors within a peer group, you can change the parameter on a peer-group basis. In this case, you do not need to remove the neighbors and change the parameter individually for each neighbor.
- If you add an outbound parameter to a peer group, that parameter is automatically applied to all neighbors

within the peer group.

- When you add a neighbor to a peer group, the software removes any outbound parameters for that neighbor from the running configuration (running-config). As a result, when you save the configuration to the startup-config file, the file does not contain any outbound parameters for the individual neighbors you have placed in a peer group. The only outbound parameters the startup-config file contains for neighbors within a peer group are the parameters associated with the peer group itself. However, the running-config and the startup-config file can contain individual parameters listed in the previous section as well as the settings for those parameters within a peer group.

You can override neighbor parameters that do not affect outbound policy on an individual neighbor basis.

- If you do not specify a parameter for an individual neighbor, the neighbor uses the value in the peer group.
- If you set the parameter for the individual neighbor, that value overrides the value you set in the peer group.
- If you add a parameter to a peer group that already contains neighbors, the parameter value is applied to neighbors that do not already have the parameter explicitly set. If a neighbor has the parameter explicitly set, the explicitly set value overrides the value you set for the peer group.
- If you remove the setting for a parameter from a peer group, the value for that parameter changes to the default value for all the neighbors in the peer group that do not have that parameter individually set.

Configuring a Peer Group

To configure a BGP4 peer group, enter commands such as the following at the BGP configuration level:

```
FastIron(config-bgp-router)#neighbor PeerGroup1 peer-group
FastIron(config-bgp-router)#neighbor PeerGroup1 description “EastCoast Neighbors”
FastIron(config-bgp-router)#neighbor PeerGroup1 remote-as 100
FastIron(config-bgp-router)#neighbor PeerGroup1 distribute-list out 1
```

The commands in this example configure a peer group called “PeerGroup1” and set the following parameters for the peer group:

- A description, “EastCoast Neighbors”
- A remote AS number, 100
- A distribute list for outbound traffic

The software applies these parameters to each neighbor you add to the peer group. You can override the description parameter for individual neighbors. If you set the description parameter for an individual neighbor, the description overrides the description configured for the peer group. However, you cannot override the remote AS and distribute list parameters for individual neighbors. Since these parameters control outbound traffic, the parameters must have the same values for all neighbors within the peer group.

Syntax: neighbor <peer-group-name> peer-group

The <peer-group-name> parameter specifies the name of the group and can be up to 80 characters long. The name can contain special characters and internal blanks. If you use internal blanks, you must use quotation marks around the name. For example, the command **neighbor “My Three Peers” peer-group** is valid, but the command **neighbor My Three Peers peer-group** is not valid.

Syntax: [no] neighbor <ip-addr> | <peer-group-name>
[advertisement-interval <num>]
[default-originate [route-map <map-name>]]
[description <string>]
[distribute-list in | out <num,num,...> | <acl-num> in | out]
[ebgp-multihop [<num>]]
[filter-list in | out <num,num,...> | <acl-num> in | out | weight]
[maximum-prefix <num> [<threshold>] [teardown]]
[next-hop-self]
[password [0 | 1] <string>]
[prefix-list <string> in | out]
[remote-as <as-number>]
[remove-private-as]
[route-map in | out <map-name>]
[route-reflector-client]
[send-community]
[soft-reconfiguration inbound]
[shutdown]
[timers keep-alive <num> hold-time <num>]
[update-source loopback <num>]
[weight <num>]

Syntax: The <ip-addr> | <peer-group-name> parameter indicates whether you are configuring a peer group or an individual neighbor. You can specify a peer group name or IP address with the **neighbor** command. If you specify a peer group name, you are configuring a peer group. If you specify a neighbor's IP address, you are configuring that individual neighbor. Use the <ip-addr> parameter if you are configuring an individual neighbor instead of a peer group. See "Adding BGP4 Neighbors" on page 38-11.

The remaining parameters are the same ones supported for individual neighbors. See "Adding BGP4 Neighbors" on page 38-11.

Applying a Peer Group to a Neighbor

After you configure a peer group, you can add neighbors to the group. When you add a neighbor to a peer group, you are applying all the neighbor attributes specified in the peer group to the neighbor.

To add neighbors to a peer group, enter commands such as the following:

```
FastIron(config-bgp-router)#neighbor 192.168.1.12 peer-group PeerGroup1
FastIron(config-bgp-router)#neighbor 192.168.2.45 peer-group PeerGroup1
FastIron(config-bgp-router)#neighbor 192.168.3.69 peer-group PeerGroup1
```

The commands in this example add three neighbors to the peer group "PeerGroup1". As members of the peer group, the neighbors automatically receive the neighbor parameter values configured for the peer group. You also can override the parameters (except parameters that govern outbound traffic) on an individual neighbor basis. For neighbor parameters not specified for the peer group, the neighbors use the default values.

Syntax: neighbor <ip-addr> peer-group <peer-group-name>

The <ip-addr> parameter specifies the IP address of the neighbor.

The <peer-group-name> parameter specifies the peer group name.

NOTE: You must add the peer group before you can add neighbors to it.

Administratively Shutting Down a Session with a BGP4 Neighbor

You can prevent the Layer 3 Switch from starting a BGP4 session with a neighbor by administratively shutting down the neighbor. This option is very useful for situations in which you want to configure parameters for a neighbor but are not ready to use the neighbor. You can shut the neighbor down as soon as you have added it the Layer 3 Switch, configure the neighbor parameters, then allow the Layer 3 Switch to re-establish a session with the neighbor by removing the shutdown option from the neighbor.

When you apply the new option to shut down a neighbor, the option takes place immediately and remains in effect until you remove the option. If you save the configuration to the startup-config file, the shutdown option remains in effect even after a software reload.

NOTE: The software also contains an option to end the session with a BGP4 neighbor and thus clear the routes learned from the neighbor. Unlike this clear option, the option for shutting down the neighbor can be saved in the startup-config file and thus can prevent the Layer 3 Switch from establishing a BGP4 session with the neighbor even after reloading the software.

NOTE: If you notice that a particular BGP4 neighbor never establishes a session with the Foundry Layer 3 Switch, check the Layer 3 Switch's running-config and startup-config files to see whether the configuration contains a command that is shutting down the neighbor. The neighbor may have been shut down previously by an administrator.

To shut down a BGP4 neighbor, enter commands such as the following:

```
FastIron(config)#router bgp
FastIron(config-bgp-router)#neighbor 209.157.22.26 shutdown
FastIron(config-bgp-router)#write memory
```

Syntax: [no] neighbor <ip-addr> shutdown

The <ip-addr> parameter specifies the IP address of the neighbor.

Optional Configuration Tasks

The following sections describe how to perform optional BGP4 configuration tasks.

Changing the Keep Alive Time and Hold Time

The Keep Alive Time specifies how frequently the router will send KEEPALIVE messages to its BGP4 neighbors. The Hold Time specifies how long the router will wait for a KEEPALIVE or UPDATE message from a neighbor before concluding that the neighbor is dead. When the router concludes that a BGP4 neighbor is dead, the router ends the BGP4 session and closes the TCP connection to the neighbor.

The default Keep Alive time is 60 seconds. The default Hold Time is 180 seconds. To change the timers, use either of the following methods.

NOTE: Generally, you should set the Hold Time to three times the value of the Keep Alive Time.

NOTE: You can override the global Keep Alive Time and Hold Time on individual neighbors. See "Adding BGP4 Neighbors" on page 38-11.

To change the Keep Alive Time to 30 and Hold Time to 90, enter the following command:

```
FastIron(config-bgp-router)#timers keep-alive 30 hold-time 90
```

Syntax: timers keep-alive <num> hold-time <num>

For each keyword, <num> indicates the number of seconds. The Keep Alive Time can be 0 – 65535. The Hold Time can be 0 or 3 – 65535 (1 and 2 are not allowed). If you set the Hold Time to 0, the router waits indefinitely for messages from a neighbor without concluding that the neighbor is dead.

Changing the BGP4 Next-Hop Update Timer

By default, the Layer 3 Switch updates its BGP4 next-hop tables and affected BGP4 routes five seconds after IGP route changes. You can change the update timer to a value from 1 – 30 seconds.

To change the BGP4 update timer value, enter a command such as the following at the BGP configuration level of the CLI:

```
FastIron(config-bgp-router)#update-time 15
```

This command changes the update timer to 15 seconds.

Syntax: [no] update-time <secs>

The <secs> parameter specifies the number of seconds and can be from 1 – 30. The default is 5.

Enabling Fast External Fallover

BGP4 routers rely on KEEPALIVE and UPDATE messages from neighbors to signify that the neighbors are alive. For BGP4 neighbors that are two or more hops away, such messages are the only indication that the BGP4 protocol has concerning the alive state of the neighbors. As a result, if a neighbor dies, the router will wait until the Hold Time expires before concluding that the neighbor is dead and closing its BGP4 session and TCP connection with the neighbor.

The router waits for the Hold Time to expire before ending the connection to a directly-attached BGP4 neighbor that dies.

For directly attached neighbors, the router immediately senses loss of a connection to the neighbor from a change of state of the port or interface that connects the router to its neighbor. For directly attached EBGP neighbors, the router can use this information to immediately close the BGP4 session and TCP connection to locally attached neighbors that die.

NOTE: The fast external fallover feature applies only to directly attached EBGP neighbors. The feature does not apply to IBGP neighbors.

If you want to enable the router to immediately close the BGP4 session and TCP connection to locally attached neighbors that die, use either of the following methods.

To enable fast external fallover, enter the following command:

```
FastIron(config-bgp-router)#fast-external-fallover
```

To disable fast external fallover again, enter the following command:

```
FastIron(config-bgp-router)#no fast-external-fallover
```

Syntax: [no] fast-external-fallover

Changing the Maximum Number of Paths for BGP4 Load Sharing

Load sharing enables the Layer 3 Switch to balance traffic to a route across multiple equal-cost paths of the same type (EBGP or IBGP) for the route.

To configure the Layer 3 Switch to perform BGP4 load sharing:

- Enable IP load sharing if it is disabled.
- Set the maximum number of paths. The default maximum number of BGP4 load sharing paths is 1, which means no BGP4 load sharing takes place by default.

  NOTE: The maximum number of BGP4 load sharing paths cannot be greater than the maximum number of IP load sharing paths.

How Load Sharing Affects Route Selection

During evaluation of multiple paths to select the best path to a given destination for installment in the IP route table, the last comparison the Layer 3 Switch performs is a comparison of the internal paths.

- When IP load sharing is disabled, the Layer 3 Switch prefers the path to the router with the lower router ID.
- When IP load sharing and BGP4 load sharing are enabled, the Layer 3 Switch balances the traffic across the multiple paths instead of choosing just one path based on router ID.

See “How BGP4 Selects a Path for a Route” on page 38-3 for a description of the BGP4 algorithm.

When you enable IP load sharing, the Layer 3 Switch can load balance BGP4 or OSPF routes across up to four equal paths by default. You can change the number of IP load sharing paths to a value from 2 – 6.

How Load Sharing Works

Load sharing is performed in round-robin fashion and is based on the destination IP address only. The first time the router receives a packet destined for a specific IP address, the router uses a round-robin algorithm to select the path that was not used for the last newly learned destination IP address. Once the router associates a path with a particular destination IP address, the router will always use that path as long as the router contains the destination IP address in its cache.

NOTE: The Layer 3 Switch does not perform source routing. The router is concerned only with the paths to the next-hop routers, not the entire paths to the destination hosts.

A BGP4 destination can be learned from multiple BGP4 neighbors, leading to multiple BGP4 paths to reach the same destination. Each of the paths may be reachable through multiple IGP paths (multiple OSPF or RIP paths). In this case, the software installs all the multiple equal-cost paths in the BGP4 route table, up to the maximum number of BGP4 equal-cost paths allowed. The IP load sharing feature then distributes traffic across the equal-cost paths to the destination.

If an IGP path used by a BGP4 next-hop route path installed in the IP route table changes, then the BGP4 paths and IP paths are adjusted accordingly. For example, if one of the OSPF paths to reach the BGP4 next hop goes down, the software removes this path from the BGP4 route table and the IP route table. Similarly, if an additional OSPF path becomes available to reach the BGP4 next-hop router for a particular destination, the software adds the additional path to the BGP4 route table and the IP route table.

Changing the Maximum Number of Shared BGP4 Paths

When IP load sharing is enabled, BGP4 can balance traffic to a specific destination across up to four equal paths. You can set the maximum number of paths to a value from 1 – 4. The default is 1.

NOTE: The maximum number of BGP4 load sharing paths cannot be greater than the maximum number of IP load sharing paths. To increase the maximum number of IP load sharing paths, use the **ip load sharing** <num> command at the global CONFIG level of the CLI.

To change the maximum number of shared paths, enter commands such as the following:

```
FastIron(config)#router bgp
FastIron(config-bgp-router)#maximum-paths 4
FastIron(config-bgp-router)#write memory
```

Syntax: [no] maximum-paths <num>

The <num> parameter specifies the maximum number of paths across which the Layer 3 Switch can balance traffic to a given BGP4 destination. You can change the maximum number of paths to a value from 2 – 4. The default is 1.

Customizing BGP4 Load Sharing

By default, when BGP4 load sharing is enabled, both IBGP and EBGP paths are eligible for load sharing, while paths from different neighboring ASs are not eligible. You can change load sharing to apply only to IBGP or EBGP paths, or to support load sharing among paths from different neighboring ASs.

To enable load sharing of IBGP paths only, enter the following command at the BGP configuration level of the CLI:

```
FastIron(config-bgp-router)#multipath ibgp
```

To enable load sharing of EBGP paths only, enter the following command at the BGP configuration level of the CLI:

```
FastIron(config-bgp-router)#multipath ebgp
```

To enable load sharing of paths from different neighboring ASs, enter the following command at the BGP configuration level of the CLI:

```
FastIron(config-bgp-router)#multipath multi-as
```

Syntax: [no] multipath ebgp | ibgp | multi-as

The **ebgp | ibgp | multi-as** parameter specifies the change you are making to load sharing:

- **ebgp** – Load sharing applies only to EBGP paths. Load sharing is disabled for IBGP paths.
- **ibgp** – Load sharing applies only to IBGP paths. Load sharing is disabled for EBGP paths.
- **multi-as** – Load sharing is enabled for paths from different ASs.

By default, load sharing applies to EBGP and IBGP paths, and does not apply to paths from different neighboring ASs.

Specifying a List of Networks to Advertise

By default, the router sends BGP4 routes only for the networks you identify using the **network** command or that are redistributed into BGP4 from RIP or OSPF. You can specify up to 600 networks.

To specify a network to be advertised, use either of the following methods.

NOTE: The exact route must exist in the IP route table before the Layer 3 Switch can create a local BGP route.

To configure the Layer 3 Switch to advertise network 209.157.22.0/24, enter the following command:

```
FastIron(config-bgp-router)#network 209.157.22.0 255.255.255.0
```

Syntax: network <ip-addr> <ip-mask> [nlri multicast | unicast | multicast unicast]
[route-map <map-name>] | [weight <num>] | [backdoor]

The <ip-addr> is the network number and the <ip-mask> specifies the network mask.

The **nlri multicast | unicast | multicast unicast** parameter specifies whether the neighbor is a multicast neighbor or a unicast neighbor. For MBGP, you must specify **multicast**. Optionally, you also can specify **unicast** if you want the Layer 3 Switch to exchange unicast (BGP4) routes as well as multicast routes with the neighbor. The default is **unicast** only.

The **route-map** <map-name> parameter specifies the name of the route map you want to use to set or change BGP4 attributes for the network you are advertising. The route map must already be configured.

The **weight** <num> parameter specifies a weight to be added to routes to this network.

The **backdoor** parameter changes the administrative distance of the route to this network from the EBGP administrative distance (20 by default) to the Local BGP weight (200 by default), thus tagging the route as a backdoor route. Use this parameter when you want the router to prefer IGP routes such as RIP or OSPF routes over the EBGP route for the network.

Specifying a Route Map Name when Configuring BGP4 Network Information

You can specify a route map as one of the parameters when you configure a BGP4 network to be advertised. The Layer 3 Switch can use the route map to set or change BGP4 attributes when creating a local BGP4 route.

To configure network information and use a route map to set or change BGP4 attributes, use the following CLI method.

NOTE: You must configure the route map before you can specify the route map name in a BGP4 network configuration.

To configure a route map, and use it to set or change route attributes for a network you define for BGP4 to advertise, enter commands such as the following:

```
FastIron(config)#route-map set_net permit 1
FastIron(config-routemap set_net)#set community no-export
FastIron(config-routemap set_net)#exit
FastIron(config)#router bgp
FastIron(config-bgp-router)#network 100.100.1.0/24 route-map set_net
```

The first two commands in this example create a route map named “set_net” that sets the community attribute for routes that use the route map to “NO_EXPORT”. The next two commands change the CLI to the BGP4 configuration level. The last command configures a network for advertising from BGP4, and associates the “set_net” route map with the network. When BGP4 originates the 100.100.1.0/24 network, BGP4 also sets the community attribute for the network to “NO_EXPORT”.

Syntax: network <ip-addr> <ip-mask> [route-map <map-name>] | [weight <num>] | [backdoor]

The **route-map** <map-name> parameter specifies the name of the route map you want to use to set or change BGP4 attributes for the network you are advertising. The route map must already be configured.

For information about the other parameters, see “Defining Route Maps” on page 38-50.

Changing the Default Local Preference

When the router uses the BGP4 algorithm to select a route to send to the IP route table, one of the parameters the algorithm uses is the local preference. Local preference is an attribute that indicates a degree of preference for a route relative to other routes. BGP4 neighbors can send the local preference value as an attribute of a route in an UPDATE message.

Local preference applies only to routes within the local AS. BGP4 routers can exchange local preference information with neighbors who also are in the local AS, but BGP4 routers do not exchange local preference information with neighbors in remote ASs.

The default local preference is 100. For routes learned from EBGP neighbors, the default local preference is assigned to learned routes. For routes learned from IBGP neighbors, the local preference value is not changed for the route.

When the BGP4 algorithm compares routes on the basis of local preferences, the route with the higher local preference is chosen.

NOTE: To set the local preference for individual routes, use route maps. See “Defining Route Maps” on page 38-50. See “How BGP4 Selects a Path for a Route” on page 38-3 for information about the BGP4 algorithm.

To change the default local preference to 200, enter the following command:

```
FastIron(config-bgp-router)#default-local-preference 200
```

Syntax: default-local-preference <num>

The <num> parameter indicates the preference and can be a value from 0 – 4294967295.

Using the IP Default Route as a Valid Next Hop for a BGP4 Route

By default, the Layer 3 Switch does not use a default route to resolve a BGP4 next-hop route. If the IP route lookup for the BGP4 next hop does not result in a valid IGP route (including static or direct routes), the BGP4 next hop is considered to be unreachable and the BGP4 route is not used.

In some cases, such as when the Layer 3 Switch is acting as an edge router, you might want to allow the device to use the default route as a valid next hop. To do so, enter the following command at the BGP4 configuration level of the CLI:

```
FastIron(config-bgp-router)#next-hop-enable-default
```

Syntax: [no] next-hop-enable-default

Advertising the Default Route

By default, the Layer 3 Switch does not originate and advertise a default route using BGP4. A BGP4 default route is the IP address 0.0.0.0 and the route prefix 0 or network mask 0.0.0.0. For example, 0.0.0.0/0 is a default route. You can enable the router to advertise a default BGP4 route using either of the following methods.

NOTE: The Foundry Layer 3 Switch checks for the existence of an IGP route for 0.0.0.0/0 in the IP route table before creating a local BGP route for 0.0.0.0/0.

To enable the router to originate and advertise a default BGP4 route, enter the following command:

```
FastIron(config-bgp-router)#default-information-originate
```

Syntax: [no] default-information-originate

Changing the Default MED (Metric) Used for Route Redistribution

The Foundry Layer 3 Switch can redistribute directly connected routes, static IP routes, RIP routes, and OSPF routes into BGP4. The MED (metric) is a global parameter that specifies the cost that will be applied to all routes by default when they are redistributed into BGP4. When routes are selected, lower metric values are preferred over higher metric values. The default BGP4 MED value is 0 and can be assigned a value from 0 – 4294967295.

NOTE: RIP and OSPF also have default metric parameters. The parameters are set independently for each protocol and have different ranges.

To change the default metric to 40, enter the following command:

```
FastIron(config-bgp-router)#default-metric 40
```

Syntax: default-metric <num>

The <num> indicates the metric and can be a value from 0 – 4294967295.

Enabling Next-Hop Recursion

For each BGP4 route a Layer 3 Switch learns, the Layer 3 Switch performs a route lookup to obtain the IP address of the route’s next hop. A BGP4 route becomes eligible for installation into the IP route table only if the following conditions are true:

- The lookup succeeds in obtaining a valid next-hop IP address for the route.
- The path to the next-hop IP address is an Interior Gateway Protocol (IGP) path or a static route path.

By default, the software performs only one lookup for a BGP route’s next-hop IP address. If the next-hop lookup does not result in a valid next-hop IP address or the path to the next-hop IP address is a BGP path, the software considers the BGP route’s destination to be unreachable. The route is not eligible to be installed in the IP route table.

It is possible for the BGP route table to contain a route whose next-hop IP address is not reachable through an IGP route, even though a hop farther away can be reached by the Layer 3 Switch through an IGP route. This can occur when the IGPs do not learn a complete set of IGP routes, resulting in the Layer 3 Switch learning about an internal route through IBGP instead of through an IGP. In this case, the IP route table does not contain a route that can be used to reach the BGP route's destination.

To enable the Layer 3 Switch to find the IGP route to a BGP route's next-hop gateway, enable recursive next-hop lookups. When you enable recursive next-hop lookup, if the first lookup for a BGP route results in an IBGP path originated within the same Autonomous System (AS), rather than an IGP path or static route path, the Layer 3 Switch performs a lookup on the next-hop gateway's next-hop IP address. If this second lookup results in an IGP path, the software considers the BGP route to be valid and thus eligible for installation in the IP route table. Otherwise, the Layer 3 Switch performs a lookup on the next-hop IP address of the next-hop gateway's next hop, and so on, until one of the lookups results in an IGP route.

NOTE: The software does not support using the default route to resolve a BGP4 route's next hop. Instead, you must configure a static route or use an IGP to learn the route to the EBGP multihop peer.

Previous software releases support use of the default route to resolve routes learned from EBGP multihop neighbors. However, even in this case Foundry recommends that you use a static route for the EBGP multihop neighbor instead. In general, we recommend that you do not use the default route as the next hop for BGP4 routes, especially when there are two or more BGP4 neighbors. Using the default route can cause loops.

Example When Recursive Route Lookups Are Disabled

Here is an example of the results of an unsuccessful next-hop lookup for a BGP route. In this case, next-hop recursive lookups are disabled. The example is for the BGP route to network 240.0.0.0/24.

```
FastIron#show ip bgp route
Total number of BGP Routes: 5
Status A:AGGREGATE B:BEST b:NOT-INSTALLED-BEST C:CONFED_EBGP D:DAMPED
       H:HISTORY I:IBGP L:LOCAL M:MULTIPATH S:SUPPRESSED
       Prefix             Next Hop        Metric     LocPrf      Weight Status
1      0.0.0.0/0           10.1.0.2       0          100         0      BI
         AS_PATH: 65001 4355 701 80
2      102.0.0.0/24        10.0.0.1       1          100         0      BI
         AS_PATH: 65001 4355 1
3      104.0.0.0/24        10.1.0.2       0          100         0      BI
         AS_PATH: 65001 4355 701 1 189
4      240.0.0.0/24       102.0.0.1       1          100         0      I
         AS_PATH: 65001 4355 3356 7170 1455
5      250.0.0.0/24       209.157.24.1    1          100         0      I
         AS_PATH: 65001 4355 701
```

In this example, the Layer 3 Switch cannot reach 240.0.0.0/24, because the next-hop IP address for the route is an IBGP route instead of an IGP route, and thus is considered unreachable by the Layer 3 Switch. Here is the IP route table entry for the BGP route's next-hop gateway (102.0.0.1/24):

```
FastIron#show ip route 102.0.0.1
Total number of IP routes: 37
      Network Address   NetMask           Gateway          Port    Cost   Type
      102.0.0.0         255.255.255.0     10.0.0.1         1/1     1      B
```

The route to the next-hop gateway is a BGP route, not an IGP route, and thus cannot be used to reach 240.0.0.0/24. In this case, the Layer 3 Switch tries to use the default route, if present, to reach the subnet that contains the BGP route's next-hop gateway.

```
FastIron#show ip route 240.0.0.0/24
Total number of IP routes: 37
      Network Address   NetMask           Gateway          Port    Cost   Type
      0.0.0.0           0.0.0.0           10.0.0.202       1/1     1      S
```

Example When Recursive Route Lookups Are Enabled

When recursive next-hop lookups are enabled, the Layer 3 Switch recursively looks up the next-hop gateways along the route until the Layer 3 Switch finds an IGP route to the BGP route's destination. Here is an example.

```
FastIron#show ip bgp route
Total number of BGP Routes: 5
Status A:AGGREGATE B:BEST b:NOT-INSTALLED-BEST C:CONFED_EBGP D:DAMPED
       H:HISTORY I:IBGP L:LOCAL M:MULTIPATH S:SUPPRESSED
       Prefix         Next Hop        Metric     LocPrf      Weight Status
1      0.0.0.0/0       10.1.0.2       0          100         0      BI
         AS_PATH: 65001 4355 701 80
2      102.0.0.0/24    10.0.0.1       1          100         0      BI
         AS_PATH: 65001 4355 1
3      104.0.0.0/24    10.1.0.2       0          100         0      BI
         AS_PATH: 65001 4355 701 1 189
4      240.0.0.0/24   102.0.0.1       1          100         0      BI
         AS_PATH: 65001 4355 3356 7170 1455
5      250.0.0.0/24   209.157.24.1    1          100         0       I
         AS_PATH: 65001 4355 701
```

The first lookup results in an IBGP route, to network 102.0.0.0/24:

```
FastIron#show ip route 102.0.0.1
Total number of IP routes: 38
      Network Address   NetMask           Gateway           Port    Cost   Type
      102.0.0.0         255.255.255.0     10.0.0.1          1/1     1      B
         AS_PATH: 65001 4355 1
```

Since the route to 102.0.0.1/24 is not an IGP route, the Layer 3 Switch cannot reach the next hop through IP, and thus cannot use the BGP route. In this case, since recursive next-hop lookups are enabled, the Layer 3 Switch next performs a lookup for 102.0.0.1's next-hop gateway, 10.0.0.1:

```
FastIron#show ip bgp route 102.0.0.0
Number of BGP Routes matching display condition : 1
Status A:AGGREGATE B:BEST b:NOT-INSTALLED-BEST C:CONFED_EBGP D:DAMPED
       H:HISTORY I:IBGP L:LOCAL M:MULTIPATH S:SUPPRESSED
       Prefix         Next Hop        Metric     LocPrf      Weight Status
1      102.0.0.0/24   10.0.0.1        1          100        0      BI
         AS_PATH: 65001 4355 1
```

The next-hop IP address for 102.0.0.1 is not an IGP route, which means the BGP route's destination still cannot be reached through IP. The recursive next-hop lookup feature performs a lookup on 10.0.0.1's next-hop gateway:

```
FastIron#show ip route 10.0.0.1
Total number of IP routes: 38
      Network Address    NetMask            Gateway            Port    Cost   Type
      10.0.0.0           255.255.255.0      0.0.0.0            1/1     1      D
        AS_PATH: 65001 4355 1
```

This lookup results in an IGP route. In fact, this route is a directly-connected route. As a result, the BGP route's destination is now reachable through IGP, which means the BGP route is eligible for installation in the IP route table. Here is the BGP route in the IP route table:

```
FastIron#show ip route 240.0.0.0/24
Total number of IP routes: 38
      Network Address     NetMask           Gateway            Port    Cost   Type
      240.0.0.0           255.255.255.0     10.0.0.1           1/1     1      B
         AS_PATH: 65001 4355 1
```

This Layer 3 Switch can use this route because the Layer 3 Switch has an IP route to the next-hop gateway. Without recursive next-hop lookups, this route would not be in the IP route table.

Enabling Recursive Next-Hop Lookups

The recursive next-hop lookups feature is disabled by default. To enable recursive next-hop lookups, enter the following command at the BGP configuration level of the CLI:

```
FastIron(config-bgp-router)#next-hop-recursion
```

Syntax: [no] next-hop-recursion

Changing Administrative Distances

BGP4 routers can learn about networks from various protocols, including the EBGP portion of BGP4 and IGPs such as OSPF and RIP. Consequently, the routes to a network may differ depending on the protocol from which the routes were learned.

To select one route over another based on the source of the route information, the Layer 3 Switch can use the administrative distances assigned to the sources. The administrative distance is a protocol-independent metric that IP routers use to compare routes from different sources.

The Layer 3 Switch re-advertises a learned best BGP4 route to the Layer 3 Switch's neighbors even when the software does not also select that route for installation in the IP route table. The best BGP4 routes is the BGP4 path that the software selects based on comparison of the paths' BGP4 route parameters. See "How BGP4 Selects a Path for a Route" on page 38-3.

When selecting a route from among different sources (BGP4, OSPF, RIP, static routes, and so on), the software compares the routes on the basis of each route's administrative distance. If the administrative distance of the paths is lower than the administrative distance of paths from other sources (such as static IP routes, RIP, or OSPF), the BGP4 paths are installed in the IP route table.

NOTE: The software will replace a statically configured default route with a learned default route if the learned route's administrative distance is lower than the statically configured default route's distance. However, the default administrative distance for static routes is changed to 1, so only directly-connected routes are preferred over static routes when the default administrative distances for the routes are used.

The following default administrative distances are found on the Foundry Layer 3 Switch:

- Directly connected – 0 (this value is not configurable)
- Static – 1 (applies to all static routes, including default routes)
- EBGP – 20
- OSPF – 110
- RIP – 120
- IBGP – 200
- Local BGP – 200
- Unknown – 255 (the router will not use this route)

Lower administrative distances are preferred over higher distances. For example, if the router receives routes for the same network from OSPF and from RIP, the router will prefer the OSPF route by default. The administrative distances are configured in different places in the software. The Layer 3 Switch re-advertises a learned best BGP4 route to neighbors by default, regardless of whether the route's administrative distance is lower than other routes from different route sources to the same destination.

- To change the EBGP, IBGP, and Local BGP default administrative distances, see the instructions in this section.
- To change the default administrative distance for OSPF, see "Modify Administrative Distance" on page 35-32.
- To change the default administrative distance for RIP, see "Changing the Administrative Distance" on page 33-5.
- To change the default administrative distance for static routes, see "Configuring Static Routes" on page 29-35.

You can change the default EBGP, IBGP, and Local BGP administrative distances using either of the following methods.

To change the default administrative distances for EBGP, IBGP, and Local BGP, enter a command such as the following:

```
FastIron(config-bgp-router)#distance 180 160 40
```

Syntax: distance <external-distance> <internal-distance> <local-distance>

The <external-distance> sets the EBGP distance and can be a value from 1 – 255.

The <internal-distance> sets the IBGP distance and can be a value from 1 – 255.

The <local-distance> sets the Local BGP distance and can be a value from 1 – 255.

Requiring the First AS to be the Neighbor's AS

By default, the Foundry device does not require the first AS listed in the AS_SEQUENCE field of an AS path Update from an EBGP neighbor to be the AS that the neighbor who sent the Update is in. You can enable the Foundry device for this requirement.

When you enable the Foundry device to require the AS that an EBGP neighbor is in to be the same as the first AS in the AS_SEQUENCE field of an Update from the neighbor, the Foundry device accepts the Update only if the ASs match. If the ASs do not match, the Foundry device sends a Notification message to the neighbor and closes the session. The requirement applies to all Updates received from EBGP neighbors.

To enable this feature, enter the following command at the BGP configuration level of the CLI:

```
FastIron(config-bgp-router)#enforce-first-as
```

Syntax: [no] enforce-first-as

Disabling or Re-Enabling Comparison of the AS-Path Length

AS-Path comparison is Step 5 in the algorithm BGP4 uses to select the next path for a route. Comparison of the AS-Path length is enabled by default. To disable it, enter the following command at the BGP configuration level of the CLI:

```
FastIron(config-bgp-router)#as-path-ignore
```

This command disables comparison of the AS-Path lengths of otherwise equal paths. When you disable AS-Path length comparison, the BGP4 algorithm shown in “How BGP4 Selects a Path for a Route” on page 38-3 skips from Step 4 to Step 6.

Syntax: [no] as-path-ignore

Enabling or Disabling Comparison of the Router IDs

Router ID comparison is Step 10 in the algorithm BGP4 uses to select the next path for a route.

NOTE: Comparison of router IDs is applicable only when BGP4 load sharing is disabled.

When router ID comparison is enabled, the path comparison algorithm compares the router IDs of the neighbors that sent the otherwise equal paths.

- If BGP4 load sharing is disabled (maximum-paths 1), the Layer 3 Switch selects the path that came from the neighbor with the lower router ID.
- If BGP4 load sharing is enabled, the Layer 3 Switch load shares among the remaining paths. In this case, the router ID is not used to select a path.

NOTE: Router ID comparison is disabled by default2. In previous releases, router ID comparison is enabled by default and cannot be disabled.

To enable router ID comparison, enter the following command at the BGP configuration level of the CLI:

```
FastIron(config-bgp-router)#compare-routerid
```

Syntax: [no] compare-routerid

For more information, see “How BGP4 Selects a Path for a Route” on page 38-3.

Configuring the Layer 3 Switch To Always Compare Multi-Exit Discriminators (MEDs)

A Multi-Exit Discriminator (MED) is a value that the BGP4 algorithm uses when comparing multiple paths received from different BGP4 neighbors in the same AS for the same route. In BGP4, a route’s MED is equivalent to its “metric”.

- BGP4 compares the MEDs of two otherwise equivalent paths ***if and only if*** the routes were learned from the same neighboring AS. This behavior is called ***deterministic MED***. 0Deterministic MED is always enabled and cannot be disabled.

  In addition, you can enable the Layer 3 Switch to always compare the MEDs, regardless of the AS information in the paths. To enable this comparison, enter the **always-compare-med** command at the BGP4 configuration level of the CLI. This option is disabled by default.
- The Layer 3 Switch compares the MEDs based on one or more of the following conditions. By default, the Layer 3 Switch compares the MEDs of paths ***only if*** the first AS in the paths is the same. (The Layer 3 Switch skips over the AS-CONFED-SEQUENCE if present.)

You can enable the Layer 3 Switch to always compare the MEDs, regardless of the AS information in the paths. For example, if the router receives UPDATES for the same route from neighbors in three ASs, the router would compare the MEDs of all the paths together, rather than comparing the MEDs for the paths in each AS individually.

NOTE: By default, value 0 (most favorable) is used in MED comparison when the MED attribute is not present. The default MED comparison results in the Layer 3 Switch favoring the route paths that are missing their MEDs. You can use the **med-missing-as-worst** command to make the Layer 3 Switch regard a BGP route with a missing MED attribute as the least favorable route, when comparing the MEDs of the routes.

NOTE: MED comparison is not performed for internal routes originated within the local AS or confederation.

To configure the router to always compare MEDs, enter the following command:

```
FastIron(config-bgp-router)#always-compare-med
```

Syntax: [no] always-compare-med

Treating Missing MEDs as the Worst MEDs

By default, the Layer 3 Switch favors a lower MED over a higher MED during MED comparison. Since the Layer 3 Switch assigns the value 0 to a route path's MED if the MED value is missing, the default MED comparison results in the Layer 3 Switch favoring the route paths that are missing their MEDs.

To change this behavior so that the Layer 3 Switch favors a route that has a MED over a route that is missing its MED, enter the following command at the BGP4 configuration level of the CLI:

```
FastIron(config-bgp-router)#med-missing-as-worst
```

Syntax: [no] med-missing-as-worst

NOTE: This command affects route selection only when route paths are selected based on MED comparison. It is still possible for a route path that is missing its MED to be selected based on other criteria. For example, a route path with no MED can be selected if its weight is larger than the weights of the other route paths.

Configuring Route Reflection Parameters

Normally, all the BGP routers within an AS are fully meshed. Each of the routers has an IBGP session with each of the other BGP routers in the AS. Each IBGP router thus has a route for each of its IBGP neighbors. For large ASs containing many IBGP routers, the IBGP route information in each of the fully-meshed IBGP routers can introduce too much administrative overhead.

To avoid this problem, you can hierarchically organize your IGP routers into clusters.

- A ***cluster*** is a group of IGP routers organized into route reflectors and route reflector clients. You configure the cluster by assigning a cluster ID on the route reflector and identifying the IGP neighbors that are members of that cluster. All the configuration for route reflection takes place on the route reflectors. The clients are unaware that they are members of a route reflection cluster. All members of the cluster must be in the same AS. The cluster ID can be any number from 1 – 4294967295. The default is the router ID, expressed as a 32-bit number.

  NOTE: If the cluster contains more than one route reflector, you need to configure the same cluster ID on all the route reflectors in the cluster. The cluster ID helps route reflectors avoid loops within the cluster.

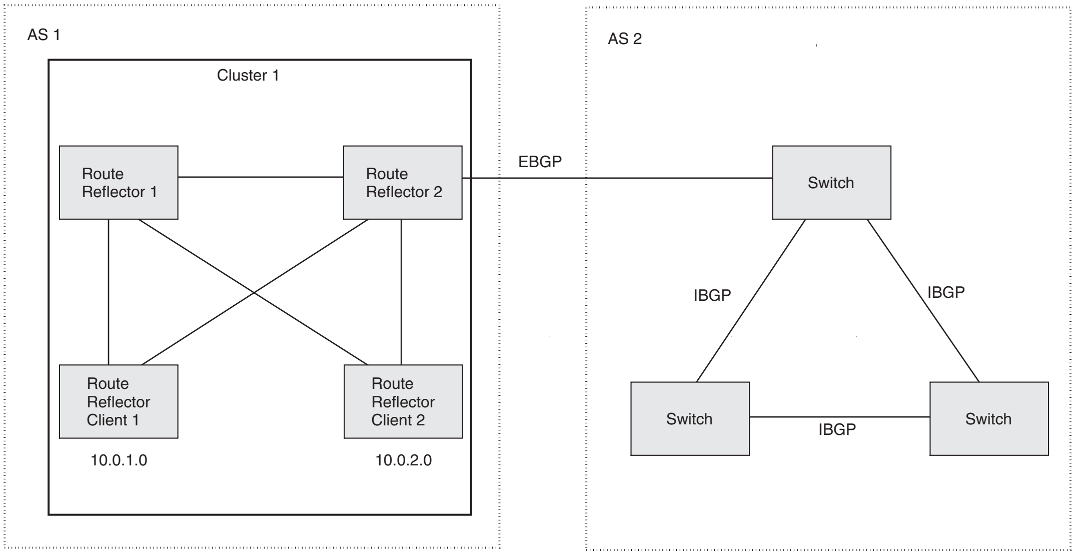
- A ***route reflector*** is an IGP router configured to send BGP route information to all the clients (other BGP4 routers) within the cluster. Route reflection is enabled on all Foundry BGP4 routers by default but does not take effect unless you add route reflector clients to the router.
- A ***route reflector client*** is an IGP router identified as a member of a cluster. You identify a router as a route reflector client on the router that is the route reflector, not on the client. The client itself requires no additional configuration. In fact, the client does not know that it is a route reflector client. The client just knows that it receives updates from its neighbors and does not know whether one or more of those neighbors are route reflectors.

NOTE: Route reflection applies only among IBGP routers within the same AS. You cannot configure a cluster that spans multiple ASs.

Figure 38.2 shows an example of a route reflector configuration. In this example, two Layer 3 Switches are configured as route reflectors for the same cluster. The route reflectors provide redundancy in case one of the reflectors becomes unavailable. Without redundancy, if a route reflector becomes unavailable, its clients are cut off from BGP4 updates.

AS1 contains a cluster with two route reflectors and two clients. The route reflectors are fully meshed with other BGP4 routers, but the clients are not fully meshed. They rely on the route reflectors to propagate BGP4 route updates.

Figure 38.2 Example of a Route Reflector Configuration



Support for RFC 2796

Route reflection on Foundry devices is based on RFC 2796. This updated RFC helps eliminate routing loops that are possible in some implementations of the older specification, RFC 1966.

NOTE: The configuration procedure for route reflection is the same regardless of whether your software release is using RFC 1966 or RFC 2796. However, the operation of the feature is different as explained below.

RFC 2796 provides more details than RFC 1966 regarding the use of the route reflection attributes, ORIGINATOR_ID and CLUSTER_LIST, to help prevent loops.

- ORIGINATOR_ID – Specifies the router ID of the BGP4 switch that originated the route. The route reflector inserts this attribute when reflecting a route to an IBGP neighbor. If a BGP4 switch receives an advertisement that contains its own router ID as the ORIGINATOR_ID, the switch discards the advertisement and does not forward it.
- CLUSTER_LIST – A list of the route reflection clusters through which the advertisement has passed. A cluster contains a route reflector and its clients. When a route reflector reflects a route, the route reflector adds its cluster ID to the front of the CLUSTER_LIST. If a route reflector receives a route that has its own cluster ID, the switch discards the advertisement and does not forward it.

The Foundry device handles the attributes as follows:

- The Layer 3 Switch adds the attributes only if it is a route reflector, and only when advertising IBGP route information to other IBGP neighbors. The attributes are not used when communicating with EBGP neighbors.

- A Layer 3 Switch configured as a route reflector sets the ORIGINATOR_ID attribute to the router ID of the router that originated the route. Moreover, the route reflector sets the attribute only if this is the first time the route is being reflected (sent by a route reflector). In previous software releases, the route reflector set the attribute to the router ID of the route reflector itself. When a Layer 3 Switch receives a route that already has the ORIGINATOR_ID attribute set, the Layer 3 Switch does not change the value of the attribute.
- If a Layer 3 Switch receives a route whose ORIGINATOR_ID attribute has the value of the Layer 3 Switch's own router ID, the Layer 3 Switch discards the route and does not advertise it. By discarding the route, the Layer 3 Switch prevents a routing loop. The Layer 3 Switch did not discard the route in previous software releases.
- The first time a route is reflected by a Layer 3 Switch configured as a route reflector, the route reflector adds the CLUSTER_LIST attribute to the route. Other route reflectors who receive the route from an IBGP neighbor add their cluster IDs to the front of the route's CLUSTER_LIST. If the route reflector does not have a cluster ID configured, the Layer 3 Switch adds its router ID to the front of the CLUSTER_LIST.
- If Layer 3 Switch configured as a route reflector receives a route whose CLUSTER_LIST contains the route reflector's own cluster ID, the route reflector discards the route and does not forward it.

Configuration Procedures

To configure a Foundry Layer 3 Switch to be a BGP4 route reflector, use either of the following methods.

NOTE: All configuration for route reflection takes place on the route reflectors, not on the clients.

Enter the following commands to configure a Foundry Layer 3 Switch as route reflector 1 in Figure 38.2 on page 38-32. To configure route reflector 2, enter the same commands on the Layer 3 Switch that will be route reflector 2. The clients require no configuration for route reflection.

```
FastIron(config-bgp-router)#cluster-id 1
FastIron(config-bgp-router)#neighbor 10.0.1.0 route-reflector-client
FastIron(config-bgp-router)#neighbor 10.0.2.0 route-reflector-client
```

Syntax: [no] cluster-id <num> | <ip-addr>

The <num> | <ip-addr> parameter specifies the cluster ID and can be a number from 1 – 4294967295 or an IP address. The default is the router ID. You can configure one cluster ID on the router. All route-reflector clients for the router are members of the cluster.

NOTE: If the cluster contains more than one route reflector, you need to configure the same cluster ID on all the route reflectors in the cluster. The cluster ID helps route reflectors avoid loops within the cluster.

To add an IBGP neighbor to the cluster, enter the following command:

Syntax: neighbor <ip-addr> route-reflector-client

For more information about the **neighbor** command, see “Adding BGP4 Neighbors” on page 38-11.

By default, the clients of a route reflector are not required to be fully meshed; the routes from a client are reflected to other clients. However, if the clients are fully meshed, route reflection is not required between clients.

If you need to disable route reflection between clients, enter the following command. When the feature is disabled, route reflection does not occur between clients but reflection does still occur between clients and non-clients.

```
FastIron(config-bgp-router)#no client-to-client-reflection
```

Enter the following command to re-enable the feature:

```
FastIron(config-bgp-router)#client-to-client-reflection
```

Syntax: [no] client-to-client-reflection

Configuring Notes

A ***confederation*** is a BGP4 Autonomous System (AS) that has been subdivided into multiple, smaller ASs. Subdividing an AS into smaller ASs simplifies administration and reduces BGP-related traffic, thus reducing the complexity of the Interior Border Gateway Protocol (IBGP) mesh among the BGP routers in the AS.

The Foundry implementation of this feature is based on RFC 3065.

Normally, all BGP routers within an AS must be fully meshed, so that each BGP router has interfaces to all the other BGP routers within the AS. This is feasible in smaller ASs but becomes unmanageable in ASs containing many BGP routers.

When you configure BGP routers into a confederation, all the routers within a sub-AS (a subdivision of the AS) use IBGP and must be fully meshed. However, routers use EBGP to communicate between different sub-ASs.

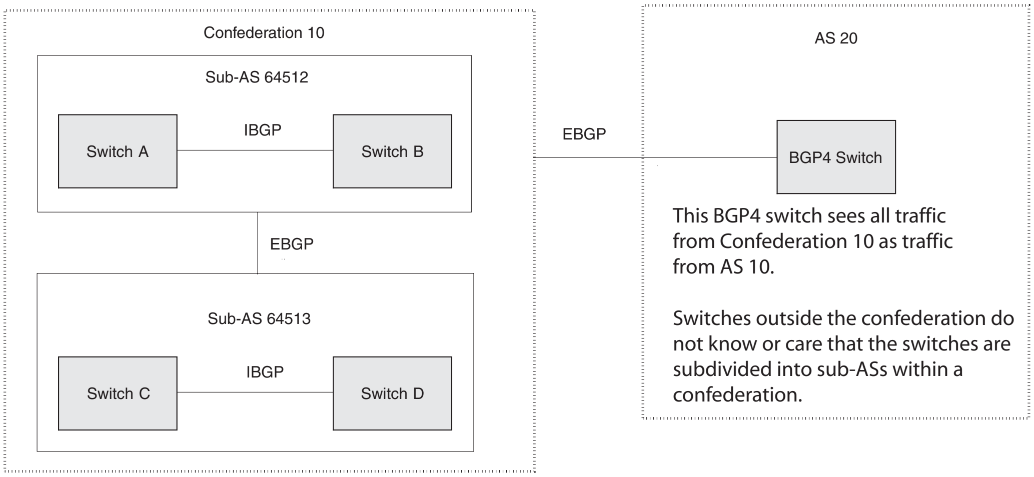
NOTE: Another method for reducing the complexity of an IBGP mesh is to use route reflection. However, if you want to run different Interior Gateway Protocols (IGPs) within an AS, configure a confederation. You can run a separate IGP within each sub-AS.

To configure a confederation, configure groups of BGP routers into sub-ASs. A sub-AS is simply an AS. The term “sub-AS” distinguishes ASs within a confederation from ASs that are not in a confederation. For the viewpoint of remote ASs, the confederation ID is the AS ID. Remote ASs do not know that the AS represents multiple sub-ASs with unique AS IDs.

NOTE: You can use any valid AS numbers for the sub-ASs. If your AS is connected to the Internet, Foundry recommends that you use numbers from within the private AS range (64512 – 65535). These are private ASs numbers and BGP4 routers do not propagate these AS numbers to the Internet.

Figure 38.3 shows an example of a BGP4 confederation.

Figure 38.3 Example of a BGP4 Confederation



This BGP4 switch sees all traffic from Confederation 10 as traffic from AS 10.

Switches outside the confederation do not know or care that the switches are subdivided into sub-ASs within a confederation.

In this example, four switches are configured into two sub-ASs, each containing two of the switches. The sub-ASs are members of confederation 10. Switches within a sub-AS must be fully meshed and communicate using IBGP. In this example, Switches A and B use IBGP to communicate. Switches C and D also use IBGP. However, the sub-ASs communicate with one another using EBGP. For example, Switch A communicates with Switch C using EBGP. The switches in the confederation communicate with other ASs using EBGP.

Switches in other ASs are unaware that Switches A – D are configured in a confederation. In fact, when switches in confederation 10 send traffic to switches in other ASs, the confederation ID is the same as the AS number for the switches in the confederation. Thus, switches in other ASs see traffic from AS 10 and are unaware that the switches in AS 10 are subdivided into sub-ASs within a confederation.

Configuring a BGP Confederation

Perform the following configuration tasks on each BGP router within the confederation:

- Configure the local AS number. The local AS number indicates membership in a sub-AS. All BGP switches with the same local AS number are members of the same sub-AS. BGP switches use the local AS number when communicating with other BGP switches within the confederation.
- Configure the confederation ID. The confederation ID is the AS number by which BGP switches outside the confederation know the confederation. Thus, a BGP switch outside the confederation is not aware and does not care that your BGP switches are in multiple sub-ASs. BGP switches use the confederation ID when communicating with switches outside the confederation. The confederation ID must be different from the sub-AS numbers.
- Configure the list of the sub-AS numbers that are members of the confederation. All the switches within the same sub-AS use IBGP to exchange switch information. Switches in different sub-ASs within the confederation use EBGP to exchange switch information.

To configure four Layer 3 Switches to be a member of confederation 10 (as shown in Figure 38.3), consisting of two sub-ASs (64512 and 64513), enter commands such as the following.

Commands for Router A

```
FastIronA(config)#router bgp
FastIronA(config-bgp-router)#local-as 64512
FastIronA(config-bgp-router)#confederation identifier 10
FastIronA(config-bgp-router)#confederation peers 64512 64513
FastIronA(config-bgp-router)#write memory
```

Syntax: local-as <num>

The <num> parameter with the **local-as** command indicates the AS number for the BGP switches within the sub-AS. You can specify a number from 1 – 65535. Foundry recommends that you use a number within the range of well-known private ASs, 64512 – 65535.

Syntax: confederation identifier <num>

The <num> parameter with the **confederation identifier** command indicates the confederation number. The confederation ID is the AS number by which BGP switches outside the confederation know the confederation. Thus, a BGP switch outside the confederation is not aware and does not care that your BGP switches are in multiple sub-ASs. BGP switches use the confederation ID when communicating with switches outside the confederation. The confederation ID must be different from the sub-AS numbers. You can specify a number from 1 – 65535.

Syntax: confederation peers <num> [<num> …]

The <num> parameter with the **confederation peers** command indicates the sub-AS numbers for the sub-ASs in the confederation. You must specify all the sub-ASs contained in the confederation. All the switches within the same sub-AS use IBGP to exchange switch information. Switches in different sub-ASs within the confederation use EBGP to exchange switch information. You can specify a number from 1 – 65535.

Commands for Router B

```
FastIronB(config)#router bgp
FastIronB(config-bgp-router)#local-as 64512
FastIronB(config-bgp-router)#confederation identifier 10
FastIronB(config-bgp-router)#confederation peers 64512 64513
FastIronB(config-bgp-router)#write memory
```

Commands for Router C

```
FastIronC(config)#router bgp
FastIronC(config-bgp-router)#local-as 64513
FastIronC(config-bgp-router)#confederation identifier 10
FastIronC(config-bgp-router)#confederation peers 64512 64513
FastIronC(config-bgp-router)#write memory
```

Commands for Router D

```
FastIronD(config)#router bgp
FastIronD(config-bgp-router)#local-as 64513
FastIronD(config-bgp-router)#confederation identifier 10
FastIronD(config-bgp-router)#confederation peers 64512 64513
FastIronD(config-bgp-router)#write memory
```

Aggregating Routes Advertised to BGP4 Neighbors

By default, the Layer 3 Switch advertises individual routes for all the networks. The aggregation feature allows you to configure the Layer 3 Switch to aggregate routes in a range of networks into a single CIDR number. For example, without aggregation, the Layer 3 Switch will individually advertise routes for networks 207.95.1.0, 207.95.2.0, and 207.95.3.0. You can configure the Layer 3 Switch to instead send a single, aggregate route for the networks. The aggregate route would be advertised as 207.95.0.0.

NOTE: To summarize CIDR networks, you must use the aggregation feature. The auto summary feature does not summarize networks that use CIDR numbers instead of class A, B, or C numbers.

To aggregate routes for 209.157.22.0, 209.157.23.0, and 209.157.24.0, enter the following command:

```
FastIron(config-bgp-router)#aggregate-address 209.157.0.0 255.255.0.0
```

Syntax: aggregate-address <ip-addr> <ip-mask> [as-set] [nlri multicast | unicast | multicast unicast] [summary-only] [suppress-map <map-name>] [advertise-map <map-name>] [attribute-map <map-name>]

The <ip-addr> and <ip-mask> parameters specify the aggregate value for the networks. Specify 0 for the host portion and for the network portion that differs among the networks in the aggregate. For example, to aggregate 10.0.1.0, 10.0.2.0, and 10.0.3.0, enter the IP address 10.0.0.0 and the network mask 255.255.0.0.

The **as-set** parameter causes the router to aggregate AS-path information for all the routes in the aggregate address into a single AS-path.

The **nlri multicast** | **unicast** | **multicast unicast** parameter specifies whether the neighbor is a multicast neighbor or a unicast neighbor. For MBGP, you must specify **multicast**. Optionally, you also can specify **unicast** if you want the Layer 3 Switch to exchange unicast (BGP4) routes as well as multicast routes with the neighbor. The default is **unicast** only.

The **summary-only** parameter prevents the router from advertising more specific routes contained within the aggregate route.

The **suppress-map** <map-name> parameter prevents the more specific routes contained in the specified route map from being advertised.

The **advertise-map** <map-name> parameter configures the router to advertise the more specific routes in the specified route map.

The **attribute-map** <map-name> parameter configures the router to set attributes for the aggregate routes based on the specified route map.

NOTE: For the **suppress-map**, **advertise-map**, and **attribute-map** parameters, the route map must already be defined. See “Defining Route Maps” on page 38-50 for information on defining a route map.

BGP Null0 Routing

Platform Support:

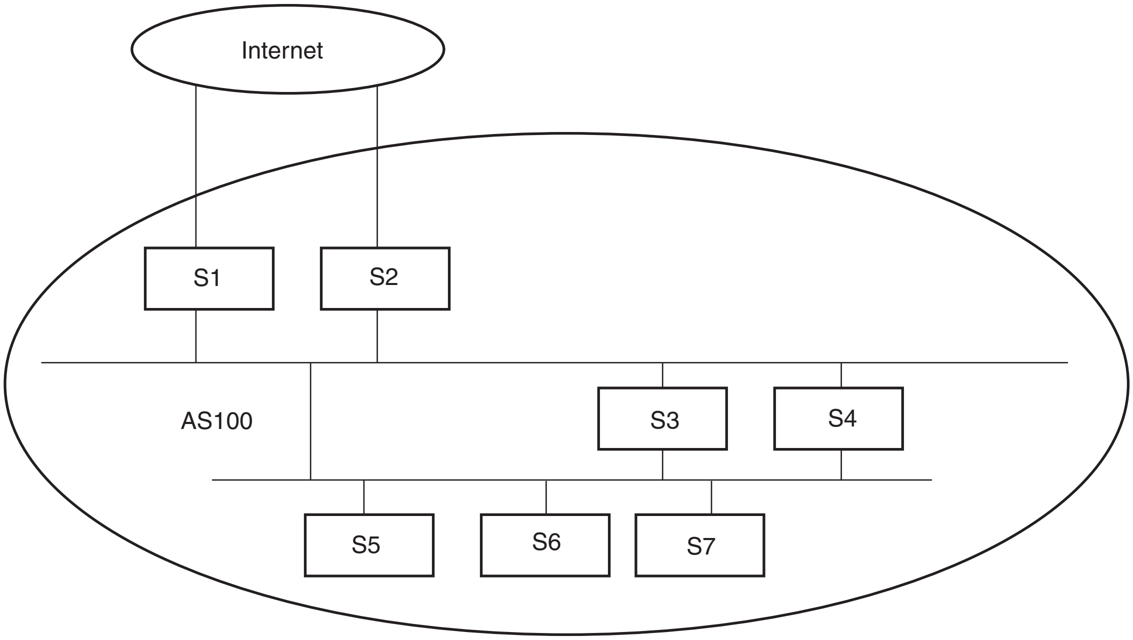
- FastIron X Series devices running software release 03.0.00 and later

The null0 routes were previously treated as invalid routes for BGP next hop resolution. BGP now uses the null0 route to resolve its next hop. Thus, null0 route in the routing table (for example, static route) is considered as a valid route by BGP. If the next hop for BGP resolves into a null0 route, the BGP route is also installed as a null0 route in the routing table.

The null0 routing feature allows network administrators to block certain network prefixes, by using null0 routes and route-maps. The combined use of null0 routes and route maps blocks traffic from a particular network prefix, telling a remote router to drop all traffic for this network prefix by redistributing a null0 route into BGP.

Figure 38.4 shows a topology for a null0 routing application example.

Figure 38.4 Example of a Null0 Routing Application



The following steps configure a null0 routing application for stopping denial of service attacks from remote hosts on the internet.

Configuration Steps

1. Select one switch, S6, to distribute null0 routes throughout the BGP network.
2. Configure a route-map to match a particular tag (50) and set the next-hop address to an unused network address (199.199.1.1).
3. Set the local-preference to a value higher than any possible internal/external local-preference (50).
4. Complete the route map by setting origin to IGP.
5. On S6, redistribute the static routes into BGP, using route-map <route-map-name> (redistribute static route-map block user).
6. On S1, the router facing the internet, configure a null0 route matching the next-hop address in the route-map (ip route 199.199.1.1/32 null0).
7. Repeat step 3 for all switches interfacing with the internet (edge corporate routers). In this case, S2 has the same null0 route as S1.
8. On S6, configure the network prefixes associated with the traffic you want to drop. The static route IP address references a destination address. You are required to point the static route to the egress port, for example, Ethernet 3/7, and specify the tag 50, matching the route-map configuration.

Configuration Examples

S6

The following configuration defines specific prefixes to filter:

```
FastIron(config)#ip route 110.0.0.40/29 ethernet 3/7 tag 50
FastIron(config)#ip route 115.0.0.192/27 ethernet 3/7 tag 50
FastIron(config)#ip route 120.014.0/23 ethernet 3/7 tag 50
```

The following configuration redistributes routes into BGP:

```
FastIron(config)#router bgp
FastIron(config-bgp-router)#local-as 100
FastIron(config-bgp-router)#neighbor <router1_int_ip address> remote-as 100
FastIron(config-bgp-router)#neighbor <router2_int_ip address> remote-as 100
FastIron(config-bgp-router)#neighbor <router3_int_ip address> remote-as 100
FastIron(config-bgp-router)#neighbor <router4_int_ip address> remote-as 100
FastIron(config-bgp-router)#neighbor <router5_int_ip address> remote-as 100
FastIron(config-bgp-router)#neighbor <router7_int_ip address> remote-as 100
FastIron(config-bgp-router)#redistribute static route-map blockuser
FastIron(config-bgp-router)#exit
```

The following configuration defines the specific next hop address and sets the local preference to preferred:

```
FastIron(config)#route-map blockuser permit 10
FastIron(config-routemap blockuser)#match tag 50
FastIron(config-routemap blockuser)#set ip next-hop 199.199.1.1
FastIron(config-routemap blockuser)#set local-preference 1000000
FastIron(config-routemap blockuser)#set origin igp
FastIron(config-routemap blockuser)#exit
```

S1

The following configuration defines the null0 route to the specific next hop address. The next hop address 199.199.1.1 points to the null0 route:

```
FastIron(config)#ip route 199.199.1.1/32 null0
FastIron(config)#router bgp
FastIron(config-bgp-router)#local-as 100
FastIron(config-bgp-router)#neighbor <router2_int_ip address> remote-as 100
FastIron(config-bgp-router)#neighbor <router3_int_ip address> remote-as 100
FastIron(config-bgp-router)#neighbor <router4_int_ip address> remote-as 100
FastIron(config-bgp-router)#neighbor <router5_int_ip address> remote-as 100
FastIron(config-bgp-router)#neighbor <router6_int_ip address> remote-as 100
FastIron(config-bgp-router)#neighbor <router7_int_ip address> remote-as 100
```

S2

The following configuration defines a null0 route to the specific next hop address. The next hop address 199.199.1.1 points to the null0 route, which gets blocked:

```
FastIron(config)#ip route 199.199.1.1/32 null0
FastIron(config)#router bgp
FastIron(config-bgp-router)#local-as 100
FastIron(config-bgp-router)#neighbor <router1_int_ip address> remote-as 100
FastIron(config-bgp-router)#neighbor <router3_int_ip address> remote-as 100
FastIron(config-bgp-router)#neighbor <router4_int_ip address> remote-as 100
FastIron (config-bgp-router)#neighbor <router5_int_ip address> remote-as 100
FastIron(config-bgp-router)#neighbor <router6_int_ip address> remote-as 100
FastIron(config-bgp-router)#neighbor <router7_int_ip address> remote-as 100
```

Show Commands

After configuring the null0 application, you can display the output:

S6

Show ip route static output for S6:

```
FastIron#show ip route static

Type Codes - B:BGP  D:Connected  S:Static  R:RIP  O:OSPF;   Cost - Dist/Metric
          Destination          Gateway         Port        Cost       Type
1       110.0.0.40/29          DIRECT          eth 3/7     1/1        S
2       115.0.0.192/27         DIRECT          eth 3/7     1/1        S
3       120.0.14.0/23          DIRECT          eth 3/7     1/1        S
```

S1 and S2

Show ip route static output for S1 and S2:

```
FastIron#show ip route static

 Type Codes - B:BGP  D:Connected  S:Static  R:RIP  O:OSPF;   Cost - Dist/Metric
             Destination        Gateway         Port        Cost       Type
1            199.199.1.1/32     DIRECT          drop        1/1        S
```

S6

Show BGP routing table output for S6

```
FastIron#show ip bgp route

Total number of BGP Routes: 126
Status A:AGGREGATE B:BEST b:NOT-INSTALLED-BEST C:CONFED_EBGP D:DAMPED  E:EBGP
H:HISTORY I:IBGP L:LOCAL M:MULTIPATH S:SUPPRESSED s:STALE
       Prefix             Next Hop        Metric     LocPrf     Weight Status
1      30.0.1.0/24        40.0.1.3        0          100        0      BI
         AS_PATH:
.             ..                       .                       .           .      .
9      110.0.0.16/30      90.0.1.3                   100        0      I
         AS_PATH: 85
10     110.0.0.40/29      199.199.1.1/32  1        1000000 32768  BL
         AS_PATH:
11     110.0.0.80/28      90.0.1.3                   100        0      I
 .           ..                       .                       .          .       .
 .           ..                       .                      .          .       .
36     115.0.0.96/28      30.0.1.3                   100        0      I
         AS_PATH: 50
37     115.0.0.192/27     199.199.1.1/32  1       10000000 32768  BL
         AS_PATH:
 .             ..                    .                         .          .      .
64     120.0.7.0/24       70.0.1.3                   100        0      I
         AS_PATH: 10
65     120.0.14.0/23      199.199.1.1/32  1        1000000 32768  BL
         AS_PATH: ..                   .                       .            .     .
```

S1 and S2

The **show ip route** output for S1 and S2 shows "drop" under the Port column for the network prefixes you configured with null0 routing:

```
FastIron#show ip route

Total number of IP routes: 133
 Type Codes - B:BGP  D:Connected  S:Static  R:RIP  O:OSPF;   Cost - Dist/Metric
        Destination        Gateway         Port        Cost       Type
1       9.0.1.24/32        DIRECT          loopback 1  0/0    D
2       30.0.1.0/24        DIRECT          eth 2/7     0/0        D
3       40.0.1.0/24        DIRECT          eth 2/1     0/0        D
.
13      110.0.0.6/31       90.0.1.3        eth 2/2     20/1       B
14      110.0.0.16/30      90.0.1.3        eth 2/2     20/1       B
15      110.0.0.40/29      DIRECT          drop        200/0      B
.          ..               .                .          .         .
42      115.0.0.192/27     DIRECT          drop        200/0      B
43      115.0.1.128/26     30.0.1.3        eth 2/7     20/1       B
.          ..               .               .           .         .
69      120.0.7.0/24       70.0.1.3        eth 2/10    20/1       B
70      120.0.14.0/23      DIRECT          drop        200/0      B
.          ..               .                .          .         .
.          ..               .                .          .         .
131     130.144.0.0/12     80.0.1.3        eth 3/4     20/1       B
132     199.199.1.1/32     DIRECT          drop        1/1        S
```

Modifying Redistribution Parameters

By default, the Layer 3 Switch does not redistribute route information between BGP4 and the IP IGPs (RIP and OSPF). You can configure the switch to redistribute OSPF routes, RIP routes, directly connected routes, or static routes into BGP4 by using the following methods.

To enable redistribution of all OSPF routes and directly attached routes into BGP4, enter the following commands.

```
FastIron(config)#router bgp
FastIron(config-bgp-router)#redistribute ospf
FastIron(config-bgp-router)#redistribute connected
FastIron(config-bgp-router)#write memory
```

Syntax: [no] redistribute connected | ospf | rip | static

The **connected** parameter indicates that you are redistributing routes to directly attached devices into BGP.

The **ospf** parameter indicates that you are redistributing OSPF routes into BGP4.

NOTE: Entering **redistribute ospf** simply redistributes internal OSPF routes. If you want to redistribute external OSPF routes also, you must use the **redistribute ospf match external...** command. See "Redistributing OSPF External Routes" on page 38-41.

The **rip** parameter indicates that you are redistributing RIP routes into BGP4.

The **static** parameter indicates that you are redistributing static routes into BGP.

See the following sections for details on redistributing specific routes using the CLI:

- "Redistributing Connected Routes" on page 38-41
- "Redistributing RIP Routes" on page 38-41

- “Redistributing OSPF External Routes” on page 38-41
- “Redistributing Static Routes” on page 38-42

Redistributing Connected Routes

To configure BGP4 to redistribute directly connected routes, enter the following command:

```
FastIron(config-bgp-router)#redistribute connected
```

Syntax: redistribute connected [metric <num>] [route-map <map-name>]

The **connected** parameter indicates that you are redistributing routes to directly attached devices into BGP4.

The **metric** <num> parameter changes the metric. You can specify a value from 0 – 4294967295. The default is 0.

The **route-map** <map-name> parameter specifies a route map to be consulted before adding the RIP route to the BGP4 route table.

NOTE: The route map you specify must already be configured on the switch. See “Defining Route Maps” on page 38-50 for information about defining route maps.

Redistributing RIP Routes

To configure BGP4 to redistribute RIP routes and add a metric of 10 to the redistributed routes, enter the following command:

```
FastIron(config-bgp-router)#redistribute rip metric 10
```

Syntax: redistribute rip [metric <num>] [route-map <map-name>]

The **rip** parameter indicates that you are redistributing RIP routes into BGP4.

The **metric** <num> parameter changes the metric. Specify a value from 0 – 4294967295. The default is 0.

The **route-map** <map-name> parameter specifies a route map to be consulted before adding the RIP route to the BGP4 route table.

NOTE: The route map you specify must already be configured on the switch. See “Defining Route Maps” on page 38-50 for information about defining route maps.

Redistributing OSPF External Routes

To configure the Layer 3 Switch to redistribute OSPF external type 1 routes, enter the following command:

```
FastIron(config-bgp-router)#redistribute ospf match external1
```

Syntax: redistribute ospf [match internal | external1 | external2] [metric <num>] [route-map <map-name>]

The **ospf** parameter indicates that you are redistributing OSPF routes into BGP4.

The **match internal | external1 | external2** parameter applies only to OSPF. This parameter specifies the types of OSPF routes to be redistributed into BGP4. The default is internal.

NOTE: If you do not enter a value for the **match** parameter, (for example, you enter r**edistribute ospf** only) then only internal OSPF routes will be redistributed.

The **metric** <num> parameter changes the metric. Specify a value from 0 – 4294967295. The default is 0.

The **route-map** <map-name> parameter specifies a route map to be consulted before adding the OSPF route to the BGP4 route table.

NOTE: The route map you specify must already be configured on the switch. See "Defining Route Maps" on page 38-50 for information about defining route maps.

NOTE: If you use both the **redistribute ospf route-map** <map-name> command and the **redistribute ospf match internal | external1 | external2** command, the software uses only the route map for filtering.

Redistributing Static Routes

To configure the Layer 3 Switch to redistribute static routes, enter the following command:

```
FastIron(config-bgp-router)#redistribute static
```

Syntax: redistribute static [metric <num>] [route-map <map-name>]

The **static** parameter indicates that you are redistributing static routes into BGP4.

The **metric** <num> parameter changes the metric. Specify a value from 0 – 4294967295. The default is 0.

The **route-map** <map-name> parameter specifies a route map to be consulted before adding the static route to the BGP4 route table.

NOTE: The route map you specify must already be configured on the switch. See "Defining Route Maps" on page 38-50 for information about defining route maps.

Disabling or Re-Enabling Re-Advertisement of All Learned BGP4 Routes to All BGP4 Neighbors

By default, the Layer 3 Switch re-advertises all learned best BGP4 routes to BGP4 neighbors, unless the routes are discarded or blocked by route maps or other filters.

If you want to prevent the Layer 3 Switch from re-advertising a learned best BGP4 route unless that route also is installed in the IP route table, use the following CLI method.

To disable re-advertisement of BGP4 routes to BGP4 neighbors except for routes that the software also installs in the route table, enter the following command:

```
FastIron(config-bgp-router)#no readvertise
```

Syntax: [no] readvertise

To re-enable re-advertisement, enter the following command:

```
FastIron(config-bgp-router)#readvertise
```

Redistributing IBGP Routes into RIP and OSPF

By default, the Layer 3 Switch does not redistribute IBGP routes from BGP4 into RIP or OSPF. This behavior helps eliminate routing loops. However, if your network can benefit from redistributing the IBGP routes from BGP4 into OSPF or RIP, you can enable the Layer 3 Switch to redistribute the routes. To do so, use the following CLI method.

To enable the Layer 3 Switch to redistribute BGP4 routes into OSPF and RIP, enter the following command:

```
FastIron(config-bgp-router)#bgp-redistribute-internal
```

Syntax: [no] bgp-redistribute-internal

To disable redistribution of IBGP routes into RIP and OSPF, enter the following command:

```
FastIron(config-bgp-router)#no bgp-redistribute-internal
```

Filtering

This section describes the following:

Filtering Specific IP Addresses

You can configure the router to explicitly permit or deny specific IP addresses received in updates from BGP4 neighbors by defining IP address filters. The router permits all IP addresses by default. You can define up to 100 IP address filters for BGP4.

- If you want ***permit*** to remain the default behavior, define individual filters to deny specific IP addresses.
- If you want to change the default behavior to ***deny***, define individual filters to permit specific IP addresses.

NOTE: Once you define a filter, the default action for addresses that do not match a filter is "deny". To change the default action to "permit", configure the last filter as "permit any any".

Address filters can be referred to by a BGP neighbor's distribute list number as well as by match statements in a route map.

NOTE: If the filter is referred to by a route map's match statement, the filter is applied in the order in which the filter is listed in the match statement.

NOTE: You also can filter on IP addresses by using IP ACLs.

To define an IP address filter to deny routes to 209.157.0.0, enter the following command:

```
FastIron(config-bgp-router)#address-filter 1 deny 209.157.0.0 255.255.0.0
```

Syntax: address-filter <num> permit | deny <ip-addr> <wildcard> <mask> <wildcard>

The <num> parameter is the filter number.

The **permit | deny** parameter indicates the action the Layer 3 Switch takes if the filter match is true.

- If you specify **permit**, the Layer 3 Switch permits the route into the BGP4 table if the filter match is true.
- If you specify **deny**, the Layer 3 Switch denies the route from entering the BGP4 table if the filter match is true.

NOTE: Once you define a filter, the default action for addresses that do not match a filter is "deny". To change the default action to "permit", configure the last filter as "permit any any".

The <ip-addr> parameter specifies the IP address. If you want the filter to match on all addresses, enter **any**.

The <wildcard> parameter specifies the portion of the IP address to match against. The <wildcard> is a four-part value in dotted-decimal notation (IP address format) consisting of ones and zeros. Zeros in the mask mean the packet's source address must match the <source-ip>. Ones mean any value matches. For example, the <ip-addr> and <wildcard> values 209.157.22.26 0.0.0.255 mean that all hosts in the Class C subnet 209.157.22.x match the policy.

If you prefer to specify the wildcard (mask value) in Classless Interdomain Routing (CIDR) format, you can enter a forward slash after the IP address, then enter the number of significant bits in the mask. For example, you can enter the CIDR equivalent of "209.157.22.26 0.0.0.255" as "209.157.22.26/24". The CLI automatically converts the CIDR number into the appropriate mask (where zeros instead of ones are the significant bits) and changes the non-significant portion of the IP address into zeros. For example, if you specify 209.157.22.26/24 or 209.157.22.26 0.0.0.255, then save the changes to the startup-config file, the value appears as 209.157.22.0/24 (if you have enabled display of subnet lengths) or 209.157.22.0 0.0.0.255 in the startup-config file.

If you enable the software to display IP subnet masks in CIDR format, the mask is saved in the file in "/<mask-bits>" format. To enable the software to display the CIDR masks, enter the **ip show-subnet-length** command at the global CONFIG level of the CLI. You can use the CIDR format to configure the filter regardless of whether the software is configured to display the masks in CIDR format.

The <mask> parameter specifies the network mask. If you want the filter to match on all destination addresses, enter **any**. The wildcard works the same as described above.

Filtering AS-Paths

You can filter updates received from BGP4 neighbors based on the contents of the AS-path list accompanying the updates. For example, if you want to deny routes that have the AS 4.3.2.1 in the AS-path from entering the BGP4 route table, you can define a filter to deny such routes.

The Layer 3 Switch provides the following methods for filtering on AS-path information:

- AS-path filters
- AS-path ACLs

NOTE: The Layer 3 Switch cannot actively support AS-path filters and AS-path ACLs at the same time. Use one method or the other but do not mix methods.

NOTE: Once you define a filter or ACL, the default action for updates that do not match a filter is "deny". To change the default action to "permit", configure the last filter or ACL as "permit any any".

AS-path filters or AS-path ACLs can be referred to by a BGP neighbor's filter list number as well as by match statements in a route map.

Defining an AS-Path Filter

To define AS-path filter 4 to permit AS 2500, enter the following command:

```
FastIron(config-bgp-router)#as-path-filter 4 permit 2500
```

Syntax: as-path-filter <num> permit | deny <as-path>

The <num> parameter identifies the filter's position in the AS-path filter list and can be from 1 – 100. Thus, the AS-path filter list can contain up to 100 filters. The Foundry Layer 3 Switch applies the filters in numerical order, beginning with the lowest-numbered filter. When a filter match is true, the Layer 3 Switch stops and does not continue applying filters from the list.

NOTE: If the filter is referred to by a route map's match statement, the filter is applied in the order in which the filter is listed in the match statement.

The **permit | deny** parameter indicates the action the router takes if the filter match is true.

- If you specify **permit**, the router permits the route into the BGP4 table if the filter match is true.
- If you specify **deny**, the router denies the route from entering the BGP4 table if the filter match is true.

The <as-path> parameter indicates the AS-path information. You can enter an exact AS-path string if you want to filter for a specific value. You also can use regular expressions in the filter string.

Defining an AS-Path ACL

To configure an AS-path list that uses ACL 1, enter a command such as the following:

```
FastIron(config)#ip as-path access-list 1 permit 100
FastIron(config)#router bgp
FastIron(config-bgp-router)#neighbor 10.10.10.1 filter-list 1 in
```

The **ip as-path** command configures an AS-path ACL that permits routes containing AS number 100 in their AS paths. The **neighbor** command then applies the AS-path ACL to advertisements and updates received from neighbor 10.10.10.1. In this example, the only routes the Layer 3 Switch permits from neighbor 10.10.10.1 are those whose AS-paths contain AS-path number 100.

Syntax: ip as-path access-list <string> [seq <seq-value>] deny | permit <regular-expression>

The <string> parameter specifies the ACL name. (If you enter a number, the CLI interprets the number as a text string.)

The **seq** <seq-value> parameter is optional and specifies the AS-path list's sequence number. You can configure up to 199 entries in an AS-path list. If you do not specify a sequence number, the software numbers them in increments of 5, beginning with number 5. The software interprets the entries in an AS-path list in numerical order, beginning with the lowest sequence number.

The **deny | permit** parameter specifies the action the software takes if a route's AS-path list matches a match statement in this ACL. To configure the AS-path match statements in a route map, use the **match as-path** command. See "Matching Based on AS-Path ACL" on page 38-53.

The <regular-expression> parameter specifies the AS path information you want to permit or deny to routes that match any of the match statements within the ACL. You can enter a specific AS number or use a regular expression. For the regular expression syntax, see "Using Regular Expressions" on page 38-45.

The **neighbor** command uses the **filter-list** parameter to apply the AS-path ACL to the neighbor. See "Adding BGP4 Neighbors" on page 38-11.

Using Regular Expressions

You use a regular expression for the <as-path> parameter to specify a single character or multiple characters as a filter pattern. If the AS-path matches the pattern specified in the regular expression, the filter evaluation is true; otherwise, the evaluation is false.

In addition, you can include special characters that influence the way the software matches the AS-path against the filter value.

To filter on a specific single-character value, enter the character for the <as-path> parameter. For example, to filter on AS-paths that contain the letter "z", enter the following command:

```
FastIron(config-bgp-router)#as-path-filter 1 permit z
```

To filter on a string of multiple characters, enter the characters in brackets. For example, to filter on AS-paths that contain "x", "y", or "z", enter the following command:

```
FastIron(config-bgp-router)#as-path-filter 1 permit [xyz]
```

Special Characters

When you enter as single-character expression or a list of characters, you also can use the following special characters. Table 38.2 on page 38-46 lists the special characters. The description for each special character includes an example. Notice that you place some special characters in front of the characters they control but you place other special characters after the characters they control. In each case, the examples show where to place the special character.

Table 38.2: BGP4 Special Characters for Regular Expressions

Character	Operation
.	The period matches on any single character, including a blank space. For example, the following regular expression matches for “aa”, “ab”, “ac”, and so on, but not just “a”. a.
*	The asterisk matches on zero or more sequences of a pattern. For example, the following regular expression matches on an AS-path that contains the string “1111” followed by any value: 1111*
+	The plus sign matches on one or more sequences of a pattern. For example, the following regular expression matches on an AS-path that contains a sequence of “g”s, such as “deg”, “degg”, “deggg”, and so on: deg+
?	The question mark matches on zero occurrences or one occurrence of a pattern. For example, the following regular expression matches on an AS-path that contains “dg” or “deg”: de?g
^	A caret (when not used within brackets) matches on the beginning of an input string. For example, the following regular expression matches on an AS-path that begins with “3”: ^3
$	A dollar sign matches on the end of an input string. For example, the following regular expression matches on an AS-path that ends with “deg”: deg$
_	An underscore matches on one or more of the following: • **,** (comma) • **{** (left curly brace) • **}** (right curly brace) • **(** (left parenthesis) • **)** (right parenthesis) • The beginning of the input string • The end of the input string • A blank space For example, the following regular expression matches on “100” but not on “1002”, “2100”, and so on. _100_

Table 38.2: BGP4 Special Characters for Regular Expressions (Continued)

Character	Operation
[ ]	Square brackets enclose a range of single-character patterns. For example, the following regular expression matches on an AS-path that contains "1", "2", "3", "4", or "5": [1-5] You can use the following expression symbols within the brackets. These symbols are allowed only inside the brackets. • **^ –** The caret matches on any characters *except* the ones in the brackets. For example, the following regular expression matches on an AS-path that does *not* contain "1", "2", "3", "4", or "5": [^1-5] • **-** The hyphen separates the beginning and ending of a range of characters. A match occurs if any of the characters within the range is present. See the example above.
|	A vertical bar (sometimes called a pipe or a "logical or") separates two alternative values or sets of values. The AS-path can match one or the other value. For example, the following regular expression matches on an AS-path that contains either "abc" or "defg": (abc)|(defg) **Note**: The parentheses group multiple characters to be treated as one value. See the following row for more information about parentheses.
( )	Parentheses allow you to create complex expressions. For example, the following complex expression matches on "abc", "abcabc", or "abcabcabcdefg", but not on "abcdefgdefg": ((abc)+)|((defg)?)

If you want to filter for a special character instead of using the special character as described in Table 38.1 on page 38-9, enter "\" (backslash) in front of the character. For example, to filter on AS-path strings containing an asterisk, enter the asterisk portion of the regular expression as "**\\***".

```
FastIron(config-bgp-router)#as-path-filter 2 deny \*
```

To use the backslash as a string character, enter two slashes. For example, to filter on AS-path strings containing a backslash, enter the backslash portion of the regular expression as "**\\**".

```
FastIron(config-bgp-router)#as-path-filter 2 deny \\
```

Filtering Communities

You can filter routes received from BGP4 neighbors based on community names. Use either of the following methods to do so.

A community is an optional attribute that identifies the route as a member of a user-defined class of routes. Community names are arbitrary values made of two five-digit integers joined by a colon. You determine what the name means when you create the community name as one of a route's attributes. Each string in the community name can be a number from 0 – 65535.

This format allows you to easily classify community names. For example, a common convention used in community naming is to configure the first string as the local AS and the second string as the unique community within that AS. Using this convention, communities 1:10, 1:20, and 1:30 can be easily identified as member communities of AS 1.

The Layer 3 Switch provides the following methods for filtering on community information:

- Community filters
- Community list ACLs

NOTE: The Layer 3 Switch cannot actively support community filters and community list ACLs at the same time. Use one method or the other but do not mix methods.

NOTE: Once you define a filter or ACL, the default action for communities that do not match a filter or ACL is "deny". To change the default action to "permit", configure the last filter or ACL entry as "permit any any".

Community filters or ACLs can be referred to by match statements in a route map.

Defining a Community Filter

To define filter 3 to permit routes that have the NO_ADVERTISE community, enter the following command:

```
FastIron(config-bgp-router)#community-filter 3 permit no-advertise
```

Syntax: community-filter <num> permit | deny <num>:<num> | internet | local-as | no-advertise | no-export

The <num> parameter identifies the filter's position in the community filter list and can be from 1 – 100. Thus, the community filter list can contain up to 100 filters. The router applies the filters in numerical order, beginning with the lowest-numbered filter. When a filter match is true, the router stops and does not continue applying filters from the list.

NOTE: If the filter is referred to by a route map's match statement, the filter is applied in the order in which the filter is listed in the match statement.

The **permit | deny** parameter indicates the action the router takes if the filter match is true.

- If you specify **permit**, the router permits the route into the BGP4 table if the filter match is true.
- If you specify **deny**, the router denies the route from entering the BGP4 table if the filter match is true.

The <num>:<num> parameter indicates a specific community number to filter. Use this parameter to filter for a private (administrator-defined) community. You can enter up to 20 community numbers with the same command.

If you want to filter for the well-known communities "LOCAL_AS", "NO_EXPORT" or "NO_ADVERTISE", use the corresponding keyword (described below).

The **internet** keyword checks for routes that do not have the community attribute. Routes without a specific community are considered by default to be members of the largest community, the Internet.

The **local-as** keyword checks for routes with the well-known community "LOCAL_AS". This community applies only to confederations. The Layer 3 Switch advertises the route only within the sub-AS. For information about confederations, see "Configuring Notes" on page 38-34.

The **no-advertise** keyword filters for routes with the well-known community "NO_ADVERTISE". A route in this community should not be advertised to any BGP4 neighbors.

The **no-export** keyword filters for routes with the well-known community "NO_EXPORT". A route in this community should not be advertised to any BGP4 neighbors outside the local AS. If the router is a member of a confederation, the Layer 3 Switch advertises the route only within the confederation. For information about confederations, see "Configuring Notes" on page 38-34.

Defining a Community ACL

To configure community ACL 1, enter a command such as the following:

```
FastIron(config)#ip community-list 1 permit 123:2
```

This command configures a community ACL that permits routes that contain community 123:2.

NOTE: See “Matching Based on Community ACL” on page 38-53 for information about how to use a community list as a match condition in a route map.

Syntax: ip community-list standard <string> [seq <seq-value>] deny | permit <community-num>

Syntax: ip community-list extended <string> [seq <seq-value>] deny | permit <community-num> | <regular-expression>

The <string> parameter specifies the ACL name. (If you enter a number, the CLI interprets the number as a text string.)

The **standard** or **extended** parameter specifies whether you are configuring a standard community ACL or an extended one. A standard community ACL does not support regular expressions whereas an extended one does. This is the only difference between standard and extended IP community lists.

The **seq** <seq-value> parameter is optional and specifies the community list’s sequence number. You can configure up to 199 entries in a community list. If you do not specify a sequence number, the software numbers them in increments of 5, beginning with number 5. The software interprets the entries in a community list in numerical order, beginning with the lowest sequence number.

The **deny** | **permit** parameter specifies the action the software takes if a route’s community list matches a match statement in this ACL. To configure the community-list match statements in a route map, use the **match community** command. See “Matching Based on Community ACL” on page 38-53.

The <community-num> parameter specifies the community type or community number. This parameter can have the following values:

- <num>**:**<num> – A specific community number
- **internet** – The Internet community
- **no-export** – The community of sub-ASs within a confederation. Routes with this community can be exported to other sub-ASs within the same confederation but cannot be exported outside the confederation to other ASs or otherwise sent to EBGP neighbors.
- **local-as** – The local sub-AS within the confederation. Routes with this community can be advertised only within the local subAS.
- **no-advertise** – Routes with this community cannot be advertised to any other BGP4 routers at all.

The <regular-expression> parameter specifies a regular expression for matching on community names. For information about regular expression syntax, see “Using Regular Expressions” on page 38-45. You can specify a regular expression only in an extended community ACL.

Defining IP Prefix Lists

An IP prefix list specifies a list of networks. When you apply an IP prefix list to a neighbor, the Layer 3 Switch sends or receives only a route whose destination is in the IP prefix list. You can configure up to 100 prefix lists. The software interprets the prefix lists in order, beginning with the lowest sequence number.

To configure an IP prefix list and apply it to a neighbor, enter commands such as the following:

```
FastIron(config)#ip prefix-list Routesfor20 permit 20.20.0.0/24
FastIron(config)#router bgp
FastIron(config-bgp-router)#neighbor 10.10.10.1 prefix-list Routesfor20 out
```

These commands configure an IP prefix list named Routesfor20, which permits routes to network 20.20.0.0/24. The **neighbor** command configures the Layer 3 Switch to use IP prefix list Routesfor20 to determine which routes to send to neighbor 10.10.10.1. The Layer 3 Switch sends routes that go to 20.20.x.x to neighbor 10.10.10.1 because the IP prefix list explicitly permits these routes to be sent to the neighbor.

Syntax: ip prefix-list <name> [seq <seq-value>] [description <string>] deny | permit <network-addr>/<mask-bits> [ge <ge-value>] [le <le-value>]

The <name> parameter specifies the prefix list name. You use this name when applying the prefix list to a neighbor.

The **description** <string> parameter is a text string describing the prefix list.

The **seq** <seq-value> parameter is optional and specifies the IP prefix list's sequence number. You can configure up to 100 prefix list entries. If you do not specify a sequence number, the software numbers them in increments of 5, beginning with prefix list entry 5. The software interprets the prefix list entries in numerical order, beginning with the lowest sequence number.

The **deny | permit** parameter specifies the action the software takes if a neighbor's route is in this prefix list.

The prefix-list matches only on this network unless you use the **ge** <ge-value> or **le** <le-value> parameters. (See below.)

The <network-addr>/<mask-bits> parameter specifies the network number and the number of bits in the network mask.

You can specify a range of prefix length for prefixes that are more specific than <network-addr>/<mask-bits>.

- If you specify only **ge** <ge-value>, then the mask-length range is from <ge-value> to 32.
- If you specify only **le** <le-value>, then the mask-length range is from length to <le-value>.

The <ge-value> or <le-value> you specify must meet the following condition:

length < ge-value <= le-value <= 32

If you do not specify **ge** <ge-value> or **le** <le-value>, the prefix list matches only on the exact network prefix you specify with the <network-addr>/<mask-bits> parameter.

For the syntax of the **neighbor** command shown in the example above, see "Adding BGP4 Neighbors" on page 38-11.

Defining Neighbor Distribute Lists

A neighbor distribute list is a list of BGP4 address filters or ACLs that filter the traffic to or from a neighbor. To configure a neighbor distribute list, use either of the following methods.

To configure a distribute list that uses ACL 1, enter a command such as the following:

```
FastIron(config-bgp-router)#neighbor 10.10.10.1 distribute-list 1 in
```

This command configures the Layer 3 Switch to use ACL 1 to select the routes that the Layer 3 Switch will accept from neighbor 10.10.10.1.

Syntax: neighbor <ip-addr> distribute-list <name-or-num> in | out

The <ip-addr> parameter specifies the neighbor.

The <name-or-num> parameter specifies the name or number of a standard, extended, or named ACL.

The **in | out** parameter specifies whether the distribute list applies to inbound or outbound routes:

- **in** – controls the routes the Layer 3 Switch will accept from the neighbor.
- **out** – controls the routes sent to the neighbor.

NOTE: The command syntax shown above is new. However, the **neighbor** <ip-addr> **distribute-list in | out** <num> command (where the direction is specified before the filter number) is the same as in earlier software releases. Use the new syntax when you are using an IP ACL with the distribute list. Use the old syntax when you are using a BGP4 address filter with the distribute list.

Defining Route Maps

A ***route map*** is a named set of match conditions and parameter settings that the router can use to modify route attributes and to control redistribution of the routes into other protocols. A route map consists of a sequence of up to 50 ***instances***. If you think of a route map as a table, an instance is a row in that table. The router evaluates a route according to a route map's instances in ascending numerical order. The route is first compared against instance 1, then against instance 2, and so on. As soon as a match is found, the router stops evaluating the route against the route map instances.

Route maps can contain ***match*** statements and ***set*** statements. Each route map contains a "permit" or "deny" action for routes that match the match statements.

- If the route map contains a permit action, a route that matches a match statement is permitted; otherwise, the route is denied.
- If the route map contains a deny action, a route that matches a match statement is denied.
- If a route does not match any match statements in the route map, the route is denied. This is the default action. To change the default action, configure the last match statement in the last instance of the route map to "permit any any".
- If there is no match statement, the software considers the route to be a match.
- For route maps that contain address filters, AS-path filters, or community filters, if the action specified by a filter conflicts with the action specified by the route map, the route map's action takes precedence over the individual filter's action.

If the route map contains set statements, routes that are permitted by the route map's match statements are modified according to the set statements.

Match statements compare the route against one or more of the following:

- The route's BGP4 MED (metric)
- A sequence of AS-path filters
- A sequence of community filters
- A sequence of address filters
- The IP address of the next hop router
- The route's tag
- For OSPF routes only, the route's type (internal, external type-1, or external type-2)
- An AS-path ACL
- A community ACL
- An IP prefix list
- An IP ACL

For routes that match all of the match statements, the route map's set statements can perform one or more of the following modifications to the route's attributes:

- Prepend AS numbers to the front of the route's AS-path. By adding AS numbers to the AS-path, you can cause the route to be less preferred when compared to other routes on the basis of the length of the AS-path.
- Add a user-defined tag to the route or add an automatically calculated tag to the route.
- Set the community value.
- Set the local preference.
- Set the MED (metric).
- Set the IP address of the next hop router.
- Set the origin to IGP or INCOMPLETE.
- Set the weight.

For example, when you configure parameters for redistributing routes into RIP, one of the optional parameters is a route map. If you specify a route map as one of the redistribution parameters, the router will match the route against the match statements in the route map. If a match is found and if the route map contains set statements, the router will set attributes in the route according to the set statements.

To create a route map, you define instances of the map. Each instance is identified by a sequence number. A route map can contain up to 50 instances.

To define a route map, use the procedures in the following sections.

Entering the Route Map Into the Software

To add instance 1 of a route map named "GET_ONE" with a permit action, enter the following command.

```
FastIron(config)#route-map GET_ONE permit 1
FastIron(config-routemap GET_ONE)#
```

Syntax: [no] route-map <map-name> permit | deny <num>

As shown in this example, the command prompt changes to the Route Map level. You can enter the match and set statements at this level. See "Specifying the Match Conditions" on page 38-52 and "Setting Parameters in the Routes" on page 38-55.

The <map-name> is a string of characters that names the map. Map names can be up to 32 characters in length.

The **permit | deny** parameter specifies the action the router will take if a route matches a match statement.

- If you specify **deny**, the Layer 3 Switch does not advertise or learn the route.
- If you specify **permit**, the Layer 3 Switch applies the match and set statements associated with this route map instance.

The <num> parameter specifies the instance of the route map you are defining. Each route map can have up to 50 instances.

To delete a route map, enter a command such as the following. When you delete a route map, all the permit and deny entries in the route map are deleted.

```
FastIron(config)#no route-map Map1
```

This command deletes a route map named "Map1". All entries in the route map are deleted.

To delete a specific instance of a route map without deleting the rest of the route map, enter a command such as the following:

```
FastIron(config)#no route-map Map1 permit 10
```

This command deletes the specified instance from the route map but leaves the other instances of the route map intact.

Specifying the Match Conditions

Use the following command to define the match conditions for instance 1 of the route map GET_ONE. This instance compares the route updates against BGP4 address filter 11.

```
FastIron(config-routemap GET_ONE)#match address-filters 11
```

Syntax: match [as-path <num>] | [address-filters | as-path-filters | community-filters <num,num,...>] | [community <num>] | [community <acl> exact-match] | [ip address <acl> | prefix-list <string>] | [ip route-source <acl> | prefix <name>] [metric <num>] | [next-hop <address-filter-list>] | [nlri multicast | unicast | multicast unicast] | [route-type internal | external-type1 | external-type2] | [tag <tag-value>]

The **as-path** <num> parameter specifies an AS-path ACL. You can specify up to five AS-path ACLs. To configure an AS-path ACL, use the **ip as-path access-list** command. See "Defining an AS-Path ACL" on page 38-45.

The **address-filters | as-path-filters | community-filters** <num,num,...> parameter specifies a filter or list of filters to be matched for each route. The router treats the first match as the best match. If a route does not match any filter in the list, then the router considers the match condition to have failed. To configure these types of filters, use commands at the BGP configuration level.

- To configure an address filter, see "Filtering Specific IP Addresses" on page 38-43.

- To configure an AS-path filter or AS-path ACL, see “Filtering AS-Paths” on page 38-44.
- To configure a community filter or community ACL, see “Filtering Communities” on page 38-47.

You can enter up to six community names on the same command line.

NOTE: The filters must already be configured.

The **community** <num> parameter specifies a community ACL.

NOTE: The ACL must already be configured.

The **community** <acl> **exact-match** parameter matches a route if (and only if) the route's community attributes field contains the same community numbers specified in the match statement.

The **ip address | next-hop** <acl-num> | prefix-list <string> parameter specifies an ACL or IP prefix list. Use this parameter to match based on the destination network or next-hop gateway. To configure an IP ACL for use with this command, use the **ip access-list** command. See “Configuring Rule-Based IP Access Control Lists (ACLs)” on page 17-1. To configure an IP prefix list, use the **ip prefix-list** command. See “Defining IP Prefix Lists” on page 38-49.

The **ip route-source** <acl> | **prefix** <name> parameter matches based on the source of a route (the IP address of the neighbor from which the Foundry device learned the route).

The **metric** <num> parameter compares the route’s MED (metric) to the specified value.

The **next-hop** <address-filter-list> parameter compares the IP address of the route’s next hop to the specified IP address filters. The filters must already be configured.

The **nlri multicast | unicast | multicast unicast** parameter specifies whether you want the route map to match on multicast routes, unicast routes, or both route types.

NOTE: By default, route maps apply to both unicast and multicast traffic.

The **route-type internal | external-type1 | external-type2** parameter applies only to OSPF routes. This parameter compares the route’s type to the specified value.

The **tag** <tag-value> parameter compares the route’s tag to the specified value.

Match Examples Using ACLs

The following sections show some detailed examples of how to configure route maps that include match statements that match on ACLs.

Matching Based on AS-Path ACL

To construct a route map that matches based on AS-path ACL 1, enter the following commands:

```
FastIron(config)#route-map PathMap permit 1
FastIron(config-routemap PathMap)#match as-path 1
```

Syntax: match as-path <num>

The <num> parameter specifies an AS-path ACL and can be a number from 1 – 199. You can specify up to five AS-path ACLs. To configure an AS-path ACL, use the **ip as-path access-list** command. See “Defining an AS-Path ACL” on page 38-45.

Matching Based on Community ACL

To construct a route map that matches based on community ACL 1, enter the following commands:

```
FastIron(config)#ip community-list 1 permit 123:2
FastIron(config)#route-map CommMap permit 1
FastIron(config-routemap CommMap)#match community 1
```

Syntax: match community <string>

The <string> parameter specifies a community list ACL. To configure a community list ACL, use the **ip community-list** command. See "Defining a Community ACL" on page 38-48.

Matching Based on Destination Network

To construct match statements for a route map that match based on destination network, use the following method. You can use the results of an IP ACL or an IP prefix list as the match condition.

```
FastIron(config)#route-map NetMap permit 1
FastIron(config-routemap NetMap)#match ip address 1
```

Syntax: match ip address <name-or-num>

Syntax: match ip address prefix-list <name>

The <name-or-num> parameter with the first command specifies an IP ACL and can be a number from 1 – 199 or the ACL name if it is a named ACL. To configure an IP ACL, use the **ip access-list** or **access-list** command. See "Configuring Rule-Based IP Access Control Lists (ACLs)" on page 17-1.

The <name> parameter with the second command specifies an IP prefix list name. To configure an IP prefix list, see "Defining IP Prefix Lists" on page 38-49.

Matching Based on Next-Hop Router

To construct match statements for a route map that match based on the IP address of the next-hop router, use either of the following methods. You can use the results of an IP ACL or an IP prefix list as the match condition.

To construct a route map that matches based on the next-hop router, enter commands such as the following:

```
FastIron(config)#route-map HopMap permit 1
FastIron(config-routemap HopMap)#match ip next-hop 2
```

Syntax: match ip next-hop <num>

Syntax: match ip next-hop prefix-list <name>

The <num> parameter with the first command specifies an IP ACL and can be a number from 1 – 199 or the ACL name if it is a named ACL. To configure an IP ACL, use the **ip access-list** or **access-list** command. See "Configuring Rule-Based IP Access Control Lists (ACLs)" on page 17-1.

The <name> parameter with the second command specifies an IP prefix list name. To configure an IP prefix list, see "Defining IP Prefix Lists" on page 38-49.

Matching Based on the Route Source

To match a BGP4 route based on its source, use the **match ip route-source** statement. Here is an example:

```
FastIron(config)#access-list 10 permit 192.168.6.0 0.0.0.255
FastIron(config)#route-map bgp1 permit 1
FastIron(config-routemap bgp1)#match ip route-source 10
```

The first command configures an IP ACL that matches on routes received from `192.168.6.0`/24. The remaining commands configure a route map that matches on all BGP4 routes advertised by the BGP4 neighbors whose addresses match addresses in the IP prefix list. You can add a set statement to change a route attribute in the routes that match. You also can use the route map as input for other commands, such as the **neighbor** and **network** commands and some show commands.

Syntax: match ip route-source <acl> | prefix <name>

The <acl> | prefix <name> parameter specifies the name or ID of an IP ACL, or an IP prefix list.

Matching On Routes Containing a Specific Set of Communities

Foundry software enables you to match routes based on the presence of a community name or number in a route, 0and to match when a route contains exactly the set of communities you specify. To match based on a set of communities, configure a community ACL that lists the communities, then compare routes against the ACL.

Here is an example.

```
FastIron(config)#ip community-list standard std_1 permit 12:34 no-export
FastIron(config)#route-map bgp2 permit 1
FastIron(config-routemap bgp2)#match community std_1 exact-match
```

The first command configures a community ACL that contains community number 12:34 and community name no-export. The remaining commands configure a route map that matches the community attributes field in BGP4 routes against the set of communities in the ACL. A route matches the route map only if the route contains all the communities in the ACL and no other communities.

Syntax: match community <acl> exact-match

The <acl> parameter specifies the name of a community list ACL. You can specify up to five ACLs. Separate the ACL names or IDs with spaces.

Here is another example.

```
FastIron(config)#ip community-list standard std_2 permit 23:45 56:78
FastIron(config)#route-map bgp3 permit 1
FastIron(config-routemap bgp3)#match community std_1 std_2 exact-match
```

These commands configure an additional community ACL, std_2, that contains community numbers 23:45 and 57:68. Route map bgp3 compares each BGP4 route against the sets of communities in ACLs std_1 and std_2. A BGP4 route that contains ***either but not both*** sets of communities matches the route map. For example, a route containing communities 23:45 and 57:68 matches. However, a route containing communities 23:45, 57:68 and 12:34, or communities 23:45, 57:68, 12:34, and no-export does not match. To match, the route's communities must be the same as those in exactly one of the community ACLs used by the match community statement.

Setting Parameters in the Routes

Use the following command to define a set statement that prepends an AS number to the AS path on each route that matches the corresponding match statement.

```
FastIron(config-routemap GET_ONE)#set as-path prepend 65535
```

Syntax: set [as-path [prepend <as-num,as-num,...>]] | [automatic-tag] | [comm-list <acl> delete] | [community <num>:<num> | <num> | internet | local-as | no-advertise | no-export] | [dampening [<half-life> <reuse> <suppress> <max-suppress-time>]] [[default] interface null0 | [ip [default] next hop <ip-addr>] [ip next-hop peer-address] | [local-preference <num>] | [metric [+ | -]<num> | none] | [metric-type type-1 | type-2] | [metric-type internal] | [next-hop <ip-addr>] | [nlri multicast | unicast | multicast unicast] | [origin igp | incomplete] | [tag <tag-value>] | [weight <num>]

The **as-path prepend** <num,num,...> parameter adds the specified AS numbers to the front of the AS-path list for the route.

The **automatic-tag** parameter calculates and sets an automatic tag value for the route.

NOTE: This parameter applies only to routes redistributed into OSPF.

The **comm-list** parameter deletes a community from a BGP4 route's community attributes field.

The **community** parameter sets the community attribute for the route to the number or well-known type you specify.

The **dampening** [<half-life> <reuse> <suppress> <max-suppress-time>] parameter sets route dampening parameters for the route. The <half-life> parameter specifies the number of minutes after which the route's penalty becomes half its value. The <reuse> parameter specifies how low a route's penalty must become before the route becomes eligible for use again after being suppressed. The <suppress> parameter specifies how high a route's penalty can become before the Layer 3 Switch suppresses the route. The <max-suppress-time> parameter specifies the maximum number of minutes that a route can be suppressed regardless of how unstable it is. For information and examples, see "Configuring Route Flap Dampening" on page 38-60.

The **[default] interface null0** parameter redirects the traffic to the specified interface. You can send the traffic to the null0 interface, which is the same as dropping the traffic. You can specify more than one interface, in which case the Layer 3 Switch uses the first available port. If the first port is unavailable, the Layer 3 Switch sends the traffic to the next port in the list. If you specify **default**, the route map redirects the traffic to the specified interface only if the Layer 3 Switch does not already have explicit routing information for the traffic. This option is used in Policy-Based Routing (PBR).

The **ip [default] next hop** <ip-addr> parameter sets the next-hop IP address for traffic that matches a match statement in the route map. If you specify **default**, the route map sets the next-hop gateway only if the Layer 3 Switch does not already have explicit routing information for the traffic. This option is used in Policy-Based Routing (PBR).

The **ip next-hop peer-address** parameter sets the BGP4 next hop for a route to the specified neighbor address.

The **local-preference** <num> parameter sets the local preference for the route. You can set the preference to a value from 0 – 4294967295.

The **metric** [**+** | **-**]<num> | none parameter sets the MED (metric) value for the route. The default MED value is 0. You can set the preference to a value from 0 – 4294967295.

- **set metric** <num> – Sets the route’s metric to the number you specify.
- **set metric +**<num> – Increases route’s metric by the number you specify.
- **set metric -**<num> – Decreases route’s metric by the number you specify.
- **set metric none** – Removes the metric from the route (removes the MED attribute from the BGP4 route).

The **metric-type type-1 | type-2** parameter changes the metric type of a route redistributed into OSPF.

The **metric-type internal** parameter sets the route's MED to the same value as the IGP metric of the BGP4 next-hop route. The parameter does this when advertising a BGP4 route to an EBGP neighbor.

The **next-hop** <ip-addr> parameter sets the IP address of the route’s next hop router.

The **nlri multicast | unicast | multicast unicast** parameter redistributes routes into the multicast Routing Information Base (RIB) instead of the unicast RIB.

NOTE: Setting the NLRI type to multicast applies only when you are using the route map to redistribute directly-connected routes. Otherwise, the set option is ignored.

The **origin igp | incomplete** parameter sets the route’s origin to IGP or INCOMPLETE.

The **tag** <tag-value> parameter sets the route’s tag. You can specify a tag value from 0 – 4294967295.

NOTE: This parameter applies only to routes redistributed into OSPF.

NOTE: You also can set the tag value using a table map. The table map changes the value only when the Layer 3 Switch places the route in the IP route table instead of changing the value in the BGP route table. See “Using a Table Map to Set the Tag Value” on page 38-57.

The **weight** <num> parameter sets the weight for the route. You can specify a weight value from 0 – 4294967295.

Setting a BP4 Route’s MED to the same Value as the IGP Metric of the Next-Hop Route

To set a route's MED to the same value as the IGP metric of the BGP4 next-hop route, when advertising the route to a neighbor, enter commands such as the following:

```
FastIron(config)#access-list 1 permit 192.168.9.0 0.0.0.255
FastIron(config)#route-map bgp4 permit 1
FastIron(config-routemap bgp4)#match ip address 1
FastIron(config-routemap bgp4)#set metric-type internal
```

The first command configures an ACL that matches on routes with destination network 192.168.9.0. The remaining commands configure a route map that matches on the destination network in ACL 1, then sets the metric type for those routes to the same value as the IGP metric of the BGP4 next-hop route.

Syntax: set metric-type internal

Setting the Next Hop of a BGP4 Route

To set the next hop address of a BGP4 route to a neighbor address, enter commands such as the following:

```
FastIron(config)#route-map bgp5 permit 1
FastIron(config-routemap bgp5)#match ip address 1
FastIron(config-routemap bgp5)#set ip next-hop peer-address
```

These commands configure a route map that matches on routes whose destination network is specified in ACL 1, and sets the next hop in the routes to the neighbor address (inbound filtering) or the local IP address of the BGP4 session (outbound filtering).

Syntax: set ip next-hop peer-address

The value that the software substitutes for **peer-address** depends on whether the route map is used for inbound filtering or outbound filtering:

- When you use the **set ip next-hop peer-address** command in an inbound route map filter, **peer-address** substitutes for the neighbor’s IP address.
- When you use the **set ip next-hop peer-address** command in an outbound route map filter, **peer-address** substitutes for the local IP address of the BGP4 session.

NOTE: You can use this command for a peer group configuration.

Deleting a Community from a BGP4 Route

To delete a community from a BGP4 route’s community attributes field, enter commands such as the following:

```
FastIron(config)#ip community-list standard std_3 permit 12:99 12:86
FastIron(config)#route-map bgp6 permit 1
FastIron(config-routemap bgp6)#match ip address 1
FastIron(config-routemap bgp6)#set comm-list std_3 delete
```

The first command configures a community ACL containing community numbers 12:99 and 12:86. The remaining commands configure a route map that matches on routes whose destination network is specified in ACL 1, and deletes communities 12:99 and 12:86 from those routes. The route does not need to contain all the specified communities in order for them to be deleted. For example, if a route contains communities 12:86, 33:44, and 66:77, community 12:86 is deleted.

Syntax: set comm-list <acl> delete

The <acl> parameter specifies the name of a community list ACL.

Using a Table Map to Set the Tag Value

Route maps that contain set statements change values in routes when the routes are accepted by the route map. For inbound route maps (route maps that filter routes received from neighbors), this means that the routes are changed before they enter the BGP4 route table.

For tag values, if you do not want the value to change until a route enters the IP route table, you can use a table map to change the value. A table map is a route map that you have associated with the IP routing table. The Layer 3 Switch applies the set statements for tag values in the table map to routes before adding them to the route table.

To configure a table map, you configure the route map, then identify it as a table map. The table map does not require separate configuration. You create it simply by calling an existing route map a table map. You can have one table map.

NOTE: Use table maps only for setting the tag value. Do not use table maps to set other attributes. To set other route attributes, use route maps or filters.

To create a route map and identify it as a table map, enter commands such as following. These commands create a route map that uses an address filter. For routes that match the address filter, the route map changes the tag value to 100. This route map is then identified as a table map. As a result, the route map is applied only to routes that the Layer 3 Switch places in the IP route table. The route map is not applied to all routes. This example assumes that address filter 11 has already been configured.

```
FastIron(config)#route-map TAG_IP permit 1
FastIron(config-routemap TAG_IP)#match address-filters 11
FastIron(config-routemap TAG_IP)#set tag 100
FastIron(config-routemap TAG_IP)#router bgp
FastIron(config-bgp-router)#table-map TAG_IP
```

Configuring Cooperative BGP4 Route Filtering

By default, the Layer 3 Switch performs all filtering of incoming routes locally, on the Layer 3 Switch itself. You can use cooperative BGP4 route filtering to cause the filtering to be performed by a neighbor before it sends the routes to the Layer 3 Switch. Cooperative filtering conserves resources by eliminating unnecessary route updates and filter processing. For example, the Layer 3 Switch can send a deny filter to its neighbor, which the neighbor uses to filter out updates before sending them to the Layer 3 Switch. The neighbor saves the resources it would otherwise use to generate the route updates, and the Layer 3 Switch saves the resources it would use to filter out the routes.

When you enable cooperative filtering, the Layer 3 Switch advertises this capability in its Open message to the neighbor when initiating the neighbor session. The Open message also indicates whether the Layer 3 Switch is configured to send filters, receive filters or both, and the types of filters it can send or receive. The Layer 3 Switch sends the filters as Outbound Route Filters (ORFs) in Route Refresh messages.

To configure cooperative filtering, perform the following tasks on the Layer 3 Switch and on its BGP4 neighbor:

- Configure the filter.

  NOTE: The current release supports cooperative filtering only for filters configured using IP prefix lists.

- Apply the filter as in ***inbound*** filter to the neighbor.
- Enable the cooperative route filtering feature on the Layer 3 Switch. You can enable the Layer 3 Switch to send ORFs to the neighbor, to receive ORFs from the neighbor, or both. The neighbor uses the ORFs you send as outbound filters when it sends routes to the Layer 3 Switch. Likewise, the Layer 3 Switch uses the ORFs it receives from the neighbor as outbound filters when sending routes to the neighbor.
- Reset the BGP4 neighbor session to send and receive ORFs.
- Perform these steps on the other device.

NOTE: If the Layer 3 Switch has inbound filters, the filters are still processed even if equivalent filters have been sent as ORFs to the neighbor.

Enabling Cooperative Filtering

To configure cooperative filtering, enter commands such as the following:

```
FastIron(config)#ip prefix-list Routesfrom1234 deny 20.20.0.0/24
FastIron(config)#ip prefix-list Routesfrom1234 permit 0.0.0.0/0 le 32
FastIron(config)#router bgp
FastIron(config-bgp-router)#neighbor 1.2.3.4 prefix-list Routesfrom1234 in
FastIron(config-bgp-router)#neighbor 1.2.3.4 capability orf prefixlist send
```

The first two commands configure statements for the IP prefix list Routesfrom1234. The first command configures a statement that denies routes to 20.20.20./24. The second command configures a statement that permits all other routes. (Once you configure an IP prefix list statement, all routes not explicitly permitted by statements in the prefix list are denied.)

The next two commands change the CLI to the BGP4 configuration level, then apply the IP prefix list to neighbor 1.2.3.4. The last command enables the Layer 3 Switch to send the IP prefix list as an ORF to neighbor 1.2.3.4. When the Layer 3 Switch sends the IP prefix list to the neighbor, the neighbor filters out the 20.20.0.x routes from its updates to the Layer 3 Switch. (This assumes that the neighbor also is configured for cooperative filtering.)

Syntax: [no] neighbor <ip-addr> | <peer-group-name> capability orf prefixlist [send | receive]

The <ip-addr> | <peer-group-name> parameter specifies the IP address of a neighbor or the name of a peer group of neighbors.

The **send** | **receive** parameter specifies the support you are enabling:

- **send** – The Layer 3 Switch sends the IP prefix lists to the neighbor.
- **receive** – The Layer 3 Switch accepts filters from the neighbor.

If you do not specify the capability, both capabilities are enabled.

The **prefixlist** parameter specifies the type of filter you want to send to the neighbor.

NOTE: The current release supports cooperative filtering only for filters configured using IP prefix lists.

Sending and Receiving ORFs

Cooperative filtering affects neighbor sessions that start after the filtering is enabled, but do not affect sessions that are already established.

To activate cooperative filtering, reset the session with the neighbor. This is required because the cooperative filtering information is exchanged in Open messages during the start of a session.

To place a prefix-list change into effect after activating cooperative filtering, perform a soft reset of the neighbor session. A soft reset does not end the current session, but sends the prefix list to the neighbor in the next route refresh message.

NOTE: Make sure cooperative filtering is enabled on the Layer 3 Switch and on the neighbor before you send the filters.

To reset a neighbor session and send ORFs to the neighbor, enter a command such as the following:

```
FastIron#clear ip bgp neighbor 1.2.3.4
```

This command resets the BGP4 session with neighbor 1.2.3.4 and sends the ORFs to the neighbor. If the neighbor sends ORFs to the Layer 3 Switch, the Layer 3 Switch accepts them if the send capability is enabled.

To perform a soft reset of a neighbor session and send ORFs to the neighbor, enter a command such as the following:

```
FastIron#clear ip bgp neighbor 1.2.3.4 soft in prefix-list
```

Syntax: clear ip bgp neighbor <ip-addr> [soft in prefix-filter]

If you use the **soft in prefix-filter** parameter, the Layer 3 Switch sends the updated IP prefix list to the neighbor as part of its route refresh message to the neighbor.

NOTE: If the Layer 3 Switch or the neighbor is not configured for cooperative filtering, the command sends a normal route refresh message.

Displaying Cooperative Filtering Information

You can display the following cooperative filtering information:

- The cooperative filtering configuration on the Layer 3 Switch.
- The ORFs received from neighbors.

To display the cooperative filtering configuration on the Layer 3 Switch, enter a command such as the following. The line shown in bold type shows the cooperative filtering status.

```
FastIron#show ip bgp neighbor 10.10.10.1
1   IP Address: 10.10.10.1, AS: 65200 (IBGP), RouterID: 10.10.10.1
    State: ESTABLISHED, Time: 0h0m7s, KeepAliveTime: 60, HoldTime: 180
       RefreshCapability: Received
       CooperativeFilteringCapability: Received
    Messages:    Open    Update   KeepAlive Notification Refresh-Req
       Sent    : 1       0        1         0            1
       Received: 1       0        1         0            1
    Last Update Time: NLRI       Withdraw          NLRI       Withdraw
                  Tx: ---        ---           Rx: ---        ---
    Last Connection Reset Reason:Unknown
    Notification Sent:     Unspecified
    Notification Received: Unspecified
    TCP Connection state: ESTABLISHED
       Byte Sent:   110, Received: 110
       Local host:  10.10.10.2, Local  Port: 8138
       Remote host: 10.10.10.1, Remote Port: 179
       ISentSeq:        460  SendNext:        571  TotUnAck:          0
       TotSent:         111  ReTrans:           0  UnAckSeq:        571
       IRcvSeq:        7349  RcvNext:        7460  SendWnd:       16384
       TotalRcv:        111  DupliRcv:          0  RcvWnd:        16384
       SendQue:           0  RcvQue:            0  CngstWnd:       5325
```

Syntax: show ip bgp neighbor <ip-addr>

To display the ORFs received from a neighbor, enter a command such as the following:

```
FastIron#show ip bgp neighbor 10.10.10.1 received prefix-filter
ip prefix-list 10.10.10.1: 4 entries
     seq 5 permit 10.10.0.0/16 ge 18 le 28
     seq 10 permit 20.20.10.0/24
     seq 15 permit 30.0.0.0/8 le 32
     seq 20 permit 40.10.0.0/16 ge 18
```

Syntax: show ip bgp neighbor <ip-addr> received prefix-filter

Configuring Route Flap Dampening

A “route flap” is the change in a route’s state, from up to down or down to up. When a route’s state changes, the state change causes changes in the route tables of the routers that support the route. Frequent changes in a route’s state can cause Internet instability and add processing overhead to the routers that support the route.

Route flap dampening is a mechanism that reduces the impact of route flap by changing a BGP4 router’s response to route state changes. When route flap dampening is configured, the Layer 3 Switch suppresses unstable routes until the route’s state changes reduce enough to meet an acceptable degree of stability. The Foundry implementation of route flap dampening is based on RFC 2439.

Route flap dampening is disabled by default. You can enable the feature globally or on an individual route basis using route maps.

NOTE: The Layer 3 Switch applies route flap dampening only to routes learned from EBGP neighbors.

The route flap dampening mechanism is based on penalties. When a route exceeds a configured penalty value, the Layer 3 Switch stops using that route and also stops advertising it to other routers. The mechanism also allows a route's penalties to reduce over time if the route's stability improves. The route flap dampening mechanism uses the following parameters:

- Suppression threshold – Specifies the penalty value at which the Layer 3 Switch stops using the route. Each time a route becomes unreachable or is withdrawn by a BGP4 UPDATE from a neighbor, the route receives a penalty of 1000. By default, when a route has a penalty value greater than 2000, the Layer 3 Switch stops using the route. Thus, by default, if a route goes down more than twice, the Layer 3 Switch stops using the route. You can set the suppression threshold to a value from 1 – 20000. The default is 2000.
- Half-life – Once a route has been assigned a penalty, the penalty decreases exponentially and decreases by half after the half-life period. The default half-life period is 15 minutes. The software reduces route penalties every five seconds. For example, if a route has a penalty of 2000 and does not receive any more penalties (it does not go down again) during the half-life, the penalty is reduced to 1000 after the half-life expires. You can configure the half-life to be from 1 – 45 minutes. The default is 15 minutes.
- Reuse threshold – Specifies the minimum penalty a route can have and still be suppressed by the Layer 3 Switch. If the route's penalty falls below this value, the Layer 3 Switch un-suppresses the route and can use it again. The software evaluates the dampened routes every ten seconds and un-suppresses the routes that have penalties below the reuse threshold. You can set the reuse threshold to a value from 1 – 20000. The default is 750.
- Maximum suppression time – Specifies the maximum number of minutes a route can be suppressed regardless of how unstable the route has been before this time. You can set the parameter to a value from 1 – 20000 minutes. The default is four times the half-life. When the half-life value is set to its default (15 minutes), the maximum suppression time defaults to 60 minutes.

You can configure route flap dampening globally or for individual routes using route maps. If you configure route flap dampening parameters globally and also use route maps, the settings in the route maps override the global values.

Globally Configuring Route Flap Dampening

To enable route flap dampening using the default values, enter the following command:

```
FastIron(config-bgp-router)#dampening
```

Syntax: dampening [<half-life> <reuse> <suppress> <max-suppress-time>]

The <half-life> parameter specifies the number of minutes after which the route's penalty becomes half its value. The route penalty allows routes that have remained stable for a while despite earlier instability to eventually become eligible for use again. The decay rate of the penalty is proportional to the value of the penalty. After the half-life expires, the penalty decays to half its value. Thus, a dampened route that is no longer unstable can eventually become eligible for use again. You can configure the half-life to be from 1 - 45 minutes. The default is 15 minutes.

The <reuse> parameter specifies how low a route's penalty must become before the route becomes eligible for use again after being suppressed. You can set the reuse threshold to a value from 1 – 20000. The default is 750 (0.75, or three-fourths, of the penalty assessed for a one "flap").

The <suppress> parameter specifies how high a route's penalty can become before the Layer 3 Switch suppresses the route. You can set the suppression threshold to a value from 1 – 20000. The default is 2000 (two "flaps").

The <max-suppress-time> parameter specifies the maximum number of minutes that a route can be suppressed regardless of how unstable it is. You can set the maximum suppression time to a value from 1 – 20000 minutes.

The default is four times the half-life setting. Thus, if you use the default half-life of 15 minutes, the maximum suppression time is 60 minutes.

The following example shows how to change the dampening parameters.

```
FastIron(config-bgp-router)#dampening 20 200 2500 40
```

This command changes the half-life to 20 minutes, the reuse threshold to 200, the suppression threshold to 2500, and the maximum number of minutes a route can be dampened to 40.

NOTE: To change any of the parameters, you must specify all the parameters with the command. If you want to leave some parameters unchanged, enter their default values.

Using a Route Map To Configure Route Flap Dampening for Specific Routes

Route maps enable you to fine tune route flap dampening parameters for individual routes. To configure route flap dampening parameters using route maps, configure BGP4 address filters for each route you want to set the dampening parameters for, then configure route map entries that set the dampening parameters for those routes. The following sections show examples.

To configure address filters and a route map for dampening specific routes, enter commands such as the following:

```
FastIron(config)#router bgp
FastIron(config-bgp-router)#address-filter 9 permit 209.157.22.0 255.255.255.0
255.255.255.0 255.255.255.0
FastIron(config-bgp-router)#address-filter 10 permit 209.157.23.0 255.255.255.0
255.255.255.0 255.255.255.0
FastIron(config-bgp-router)#exit
FastIron(config)#route-map DAMPENING_MAP permit 9
FastIron(config-routemap DAMPENING_MAP)#match address-filters 9
FastIron(config-routemap DAMPENING_MAP)#set dampening 10 200 2500 40
FastIron(config-routemap DAMPENING_MAP)#exit
FastIron(config)#route-map DAMPENING_MAP permit 10
FastIron(config-routemap DAMPENING_MAP)#match address-filters 10
FastIron(config-routemap DAMPENING_MAP)#set dampening 20 200 2500 60
FastIron(config-routemap DAMPENING_MAP)#router bgp
FastIron(config-bgp-router)#dampening route-map DAMPENING_MAP
```

The **address-filter** commands in this example configure two BGP4 address filters, for networks 209.157.22.0 and 209.157.23.0. The first route-map command creates an entry in a route map called “DAMPENING_MAP”. Within this entry of the route map, the **match** command matches based on address filter 9, and the **set** command sets the dampening parameters for the route that matches. Thus, for BGP4 routes to 209.157.22.0, the Layer 3 Switch uses the route map to set the dampening parameters. These parameters override the globally configured dampening parameters.

The commands for the second entry in the route map (instance 10 in this example) perform the same functions for route 209.157.23.0. Notice that the dampening parameters are different for each route.

Using a Route Map To Configure Route Flap Dampening for a Specific Neighbor

You can use a route map to configure route flap dampening for a specific neighbor by performing the following tasks:

- Configure an empty route map with no match or set statements. This route map does not specify particular routes for dampening but does allow you to enable dampening globally when you refer to this route map from within the BGP configuration level.
- Configure another route map that explicitly enables dampening. Use a set statement within the route map to

enable dampening. When you associate this route map with a specific neighbor, the route map enables dampening for all routes associated with the neighbor. You also can use match statements within the route map to selectively perform dampening on some routes from the neighbor.

NOTE: You still need to configure the first route map to enable dampening globally. The second route map does not enable dampening by itself; it just applies dampening to a neighbor.

- Apply the route map to the neighbor.

To enable route flap dampening for a specific BGP4 neighbor, enter commands such as the following:

```
FastIron(config)#route-map DAMPENING_MAP_ENABLE permit 1
FastIron(config-routemap DAMPENING_MAP_ENABLE)#exit
FastIron(config)#route-map DAMPENING_MAP_NEIGHBOR_A permit 1
FastIron(config-routemap DAMPENING_MAP_NEIGHBOR_A)#set dampening
FastIron(config-routemap DAMPENING_MAP_NEIGHBOR_A)#exit
FastIron(config)#router bgp
FastIron(config-bgp-router)#dampening route-map DAMPENING_MAP_ENABLE
FastIron(config-bgp-router)#neighbor 10.10.10.1 route-map in
DAMPENING_MAP_NEIGHBOR_A
```

In this example, the first command globally enables route flap dampening. This route map does not contain any match or set statements. At the BGP configuration level, the **dampening route-map** command refers to the DAMPENING_MAP_ENABLE route map created by the first command, thus enabling dampening globally.

The third and fourth commands configure a second route map that explicitly enables dampening. Notice that the route map does not contain a match statement. The route map implicitly applies to all routes. Since the route map will be applied to a neighbor at the BGP configuration level, the route map will apply to all routes associated with the neighbor.

Although the second route map enables dampening, the first route map is still required. The second route map enables dampening for the neighbors to which the route map is applied. However, unless dampening is already enabled globally by the first route map, the second route map has no effect.

The last two commands apply the route maps. The **dampening route-map** command applies the first route map, which enables dampening globally. The **neighbor** command applies the second route map to neighbor 10.10.10.1. Since the second route map does not contain match statements for specific routes, the route map enables dampening for all routes received from the neighbor.

Removing Route Dampening from a Route

You can un-suppress routes by removing route flap dampening from the routes. The Layer 3 Switch allows you to un-suppress all routes at once or un-suppress individual routes.

To un-suppress all the suppressed routes, enter the following command at the Privileged EXEC level of the CLI:

```
FastIron#clear ip bgp damping
```

Syntax: clear ip bgp damping [<ip-addr> <ip-mask>]

The <ip-addr> parameter specifies a particular network.

The <ip-mask> parameter specifies the network's mask.

To un-suppress a specific route, enter a command such as the following:

```
FastIron#clear ip bgp damping 209.157.22.0 255.255.255.0
```

This command un-suppresses only the route(s) for network 209.157.22.0/24.

Removing Route Dampening from a Neighbor's Routes Suppressed Due to Aggregation

You can selectively unsuppress more-specific routes that have been suppressed due to aggregation, and allow the routes to be advertised to a specific neighbor or peer group.

Here is an example.

```
FastIron(config-bgp-router)#aggregate-address 209.1.0.0 255.255.0.0 summary-only
FastIron(config-bgp-router)#show ip bgp route 209.1.0.0/16 longer
Number of BGP Routes matching display condition : 2
Status A:AGGREGATE B:BEST b:NOT-INSTALLED-BEST C:CONFED_EBGP D:DAMPED
       E:EBGP H:HISTORY I:IBGP L:LOCAL M:MULTIPATH S:SUPPRESSED F:FILTERED
       Prefix             Next Hop        Metric     LocPrf     Weight Status
1      209.1.0.0/16       0.0.0.0                    101        32768  BAL
         AS_PATH:
2      209.1.44.0/24      10.2.0.1        1          101        32768  BLS
         AS_PATH:
```

The **aggregate-address** command configures an aggregate address. The **summary-only** parameter prevents the Layer 3 Switch from advertising more specific routes contained within the aggregate route. The **show ip bgp route** command shows that the more specific routes aggregated into 209.1.0.0/16 have been suppressed. In this case, the route to 209.1.44.0/24 has been suppressed. The following command indicates that the route is not being advertised to the Layer 3 Switch's BGP4 neighbors.

```
FastIron#show ip bgp route 209.1.44.0/24
Number of BGP Routes matching display condition : 1
Status A:AGGREGATE B:BEST b:NOT-INSTALLED-BEST C:CONFED_EBGP D:DAMPED
       E:EBGP H:HISTORY I:IBGP L:LOCAL M:MULTIPATH S:SUPPRESSED F:FILTERED
       Prefix             Next Hop        Metric     LocPrf     Weight Status
1      209.1.44.0/24      10.2.0.1        1          101        32768  BLS
         AS_PATH:
       Route is not advertised to any peers
```

If you want to override the **summary-only** parameter and allow a specific route to be advertised to a neighbor, enter commands such as the following:

```
FastIron(config)#ip prefix-list Unsuppress1 permit 209.1.44.0/24
FastIron(config)#route-map RouteMap1 permit 1
FastIron(config-routemap RouteMap1)#match prefix-list Unsuppress1
FastIron(config-routemap RouteMap1)#exit
FastIron(config)#router bgp
FastIron(config-bgp-router)#neighbor 10.1.0.2 unsuppress-map RouteMap1
FastIron(config-bgp-router)#clear ip bgp neighbor 10.1.0.2 soft-out
```

The **ip prefix-list** command configures an IP prefix list for network 209.1.44.0/24, which is the route you want to unsuppress. The next two commands configure a route map that uses the prefix list as input. The **neighbor** command enables the Layer 3 Switch to advertise the routes specified in the route map to neighbor 10.1.0.2. The **clear** command performs a soft reset of the session with the neighbor so that the Layer 3 Switch can advertise the unsuppressed route.

Syntax: [no] neighbor <ip-addr> | <peer-group-name> unsuppress-map <map-name>

The following command verifies that the route has been unsuppressed.

```
FastIron#show ip bgp route 209.1.44.0/24
Number of BGP Routes matching display condition : 1
Status A:AGGREGATE B:BEST b:NOT-INSTALLED-BEST C:CONFED_EBGP D:DAMPED
       E:EBGP H:HISTORY I:IBGP L:LOCAL M:MULTIPATH S:SUPPRESSED F:FILTERED
       Prefix             Next Hop        Metric     LocPrf     Weight Status
1      209.1.44.0/24      10.2.0.1        1          101        32768  BLS
         AS_PATH:
       Route is advertised to 1 peers:
        10.1.0.2(4)
```

Displaying and Clearing Route Flap Dampening Statistics

The software provides many options for displaying and clearing route flap statistics. To display the statistics, use either of the following methods.

Displaying Route Flap Dampening Statistics

To display route dampening statistics or all the dampened routes, enter the following command at any level of the CLI:

```
FastIron#show ip bgp flap-statistics

Total number of flapping routes: 414
    Status Code  >:best d:damped h:history *:valid
    Network            From             Flaps Since    Reuse    Path
h>  192.50.206.0/23    166.90.213.77    1     0 :0 :13 0 :0 :0  65001 4355 1 701
h>  203.255.192.0/20   166.90.213.77    1     0 :0 :13 0 :0 :0  65001 4355 1 7018
h>  203.252.165.0/24   166.90.213.77    1     0 :0 :13 0 :0 :0  65001 4355 1 7018
h>  192.50.208.0/23    166.90.213.77    1     0 :0 :13 0 :0 :0  65001 4355 1 701
h>  133.33.0.0/16      166.90.213.77    1     0 :0 :13 0 :0 :0  65001 4355 1 701
*>  204.17.220.0/24    166.90.213.77    1     0 :1 :4  0 :0 :0  65001 4355 701 62
```

Syntax: show ip bgp flap-statistics [regular-expression <regular-expression> | <address> <mask> [longer-prefixes] | neighbor <ip-addr>]

The **regular-expression** <regular-expression> parameter is a regular expression. The regular expressions are the same ones supported for BGP4 AS-path filters. See “Using Regular Expressions” on page 38-45.

The <address> <mask> parameter specifies a particular route. If you also use the optional **longer-prefixes** parameter, then all statistics for routes that match the specified route or have a longer prefix than the specified route are displayed. For example, if you specify **209.157.0.0 longer**, then all routes with the prefix 209.157. or that have a longer prefix (such as 209.157.22.) are displayed.

The **neighbor** <ip-addr> parameter displays route flap dampening statistics only for routes learned from the specified neighbor. You also can display route flap statistics for routes learned from a neighbor by entering the following command: **show ip bgp neighbor** <ip-addr> **flap-statistics**.

This display shows the following information.

Table 38.3: Route Flap Dampening Statistics

This Field...	Displays...
Total number of flapping routes	Total number of routes in the Layer 3 Switch’s BGP4 route table that have changed state and thus have been marked as flapping routes.

Table 38.3: Route Flap Dampening Statistics

This Field...	Displays...
Status code	Indicates the dampening status of the route, which can be one of the following: • > – This is the best route among those in the BGP4 route table to the route's destination. • d – This route is currently dampened, and thus unusable. • h – The route has a history of flapping and is unreachable now. • * – The route has a history of flapping but is currently usable.
Network	The destination network of the route.
From	The neighbor that sent the route to the Layer 3 Switch.
Flaps	The number of flaps (state changes) the route has experienced.
Since	The amount of time since the first flap of this route.
Reuse	The amount of time remaining until this route will be un-suppressed and thus be usable again.
Path	Shows the AS-path information for the route.

You also can display all the dampened routes by entering the following command: **show ip bgp dampened-paths**.

Clearing Route Flap Dampening Statistics

To clear route flap dampening statistics, use the following CLI method.

NOTE: Clearing the dampening statistics for a route does not change the dampening status of the route.

To clear all the route dampening statistics, enter the following command at any level of the CLI:

```
FastIron#clear ip bgp flap-statistics
```

Syntax: clear ip bgp flap-statistics [regular-expression <regular-expression> | <address> <mask> | neighbor <ip-addr>]

The parameters are the same as those for the **show ip bgp flap-statistics** command (except the **longer-prefixes** option is not supported). See "Displaying Route Flap Dampening Statistics" on page 38-65.

NOTE: The **clear ip bgp damping** command not only clears statistics but also un-suppresses the routes. See "Displaying Route Flap Dampening Statistics" on page 38-65.

Generating Traps for BGP

You can enable and disable SNMP traps for BGP. BGP traps are enabled by default.

To enable BGP traps after they have been disabled, enter the following command:

```
FastIron(config)#snmp-server enable traps bgp
```

Syntax: [no] snmp-server enable traps bgp

Use the **no** form of the command to disable BGP traps.

Displaying BGP4 Information

You can display the following configuration information and statistics for the BGP4 protocol on the router:

- Summary BGP4 configuration information for the router
- Active BGP4 configuration information (the BGP4 information in the running-config)
- CPU utilization statistics
- Neighbor information
- Peer-group information
- Information about the paths from which BGP4 selects routes
- Summary BGP4 route information
- The router’s BGP4 route table
- Route flap dampening statistics
- Active route maps (the route map configuration information in the running-config)

Displaying Summary BGP4 Information

You can display the local AS number, the maximum number of routes and neighbors supported, and some BGP4 statistics.

To view summary BGP4 information for the router, enter the following command at any CLI prompt:

```
FastIron#show ip bgp summary
  BGP4 Summary
  Router ID: 101.0.0.1   Local AS Number : 4
  Confederation Identifier : not configured
  Confederation Peers: 4 5
  Maximum Number of Paths Supported for Load Sharing : 1
  Number of Neighbors Configured : 11
  Number of Routes Installed : 2
  Number of Routes Advertising to All Neighbors : 8
  Number of Attribute Entries Installed : 6
  Neighbor Address  AS#  State   Time     Rt:Accepted Filtered Sent   ToSend
  1.2.3.4           200   ADMDN   0h44m56s    0        0        0      2
  10.0.0.2          5     ADMDN   0h44m56s    0        0        0      0
  10.1.0.2          5     ESTAB   0h44m56s    1        11       0      0
  10.2.0.2          5     ESTAB   0h44m55s    1        0        0      0
  10.3.0.2          5     ADMDN   0h25m28s    0        0        0      0
  10.4.0.2          5     ADMDN   0h25m31s    0        0        0      0
  10.5.0.2          5     CONN    0h 0m 8s    0        0        0      0
  10.7.0.2          5     ADMDN   0h44m56s    0        0        0      0
  100.0.0.1         4     ADMDN   0h44m56s    0        0        0      2
  102.0.0.1         4     ADMDN   0h44m56s    0        0        0      2
  150.150.150.150   0     ADMDN   0h44m56s    0        0        0      2
```

This display shows the following information.

Table 38.4: BGP4 Summary Information

This Field...	Displays...
Router ID	The Layer 3 Switch’s router ID.

Table 38.4: BGP4 Summary Information (Continued)

This Field...	Displays...
Local AS Number	The BGP4 AS number the router is in.
Confederation Identifier	The AS number of the confederation the Layer 3 Switch is in.
Confederation Peers	The numbers of the local ASs contained in the confederation. This list matches the confederation peer list you configure on the Layer 3 Switch.
Maximum Number of Paths Supported for Load Sharing	The maximum number of route paths across which the device can balance traffic to the same destination. The feature is enabled by default but the default number of paths is 1. You can increase the number from 2 – 4 paths. See “Changing the Maximum Number of Paths for BGP4 Load Sharing” on page 38-21.
Number of Neighbors Configured	The number of BGP4 neighbors configured on this Layer 3 Switch.
Number of Routes Installed	The number of BGP4 routes in the router’s BGP4 route table. To display the BGP4 route table, see “Displaying the BGP4 Route Table” on page 38-90.
Number of Routes Advertising to All Neighbors	The total of the RtSent and RtToSend columns for all neighbors.
Number of Attribute Entries Installed	The number of BGP4 route-attribute entries in the router’s route-attributes table. To display the route-attribute table, see “Displaying BGP4 Route-Attribute Entries” on page 38-98.
Neighbor Address	The IP addresses of this router’s BGP4 neighbors.
AS#	The AS number.

Table 38.4: BGP4 Summary Information (Continued)

This Field...	Displays...
State	The state of this router's neighbor session with each neighbor. The states are from this router's perspective of the session, not the neighbor's perspective. The state values are based on the BGP4 state machine values described in RFC 1771 and can be one of the following for each router: • IDLE – The BGP4 process is waiting to be started. Usually, enabling BGP4 or establishing a neighbor session starts the BGP4 process. • A minus sign (-) indicates that the session has gone down and the software is clearing or removing routes. • ADMND – The neighbor has been administratively shut down. See "Administratively Shutting Down a Session with a BGP4 Neighbor" on page 38-19. • A minus sign (-) indicates that the session has gone down and the software is clearing or removing routes. • CONNECT – BGP4 is waiting for the connection process for the TCP neighbor session to be completed. • ACTIVE – BGP4 is waiting for a TCP connection from the neighbor. **Note**: If the state frequently changes between CONNECT and ACTIVE, there may be a problem with the TCP connection. • OPEN SENT – BGP4 is waiting for an Open message from the neighbor. • OPEN CONFIRM – BGP4 has received an OPEN message from the neighbor and is now waiting for either a KEEPALIVE or NOTIFICATION message. If the router receives a KEEPALIVE message from the neighbor, the state changes to Established. If the message is a NOTIFICATION, the state changes to Idle. • ESTABLISHED – BGP4 is ready to exchange UPDATE packets with the neighbor. • If there is more BGP data in the TCP receiver queue, a plus sign (+) is also displayed. **Note**: If you display information for the neighbor using the **show ip bgp neighbor** <ip-addr> command, the TCP receiver queue value will be greater than 0.
Time	The time that has passed since the state last changed.
Accepted	The number of routes received from the neighbor that this router installed in the BGP4 route table. Usually, this number is lower than the RoutesRcvd number. The difference indicates that this router filtered out some of the routes received in the UPDATE messages.

Table 38.4: BGP4 Summary Information (Continued)

This Field...	Displays...
Filtered	The routes or prefixes that have been filtered out. • If soft reconfiguration is enabled, this field shows how many routes were filtered out (not placed in the BGP4 route table) but retained in memory. • If soft reconfiguration is not enabled, this field shows the number of BGP4 routes that have been filtered out.
Sent	The number of BGP4 routes that the Layer 3 Switch has sent to the neighbor.
ToSend	The number of routes the Layer 3 Switch has queued to send to this neighbor.

Displaying the Active BGP4 Configuration

To view the active BGP4 configuration information contained in the running-config without displaying the entire running-config, use the following CLI method.

To display the device's active BGP4 configuration, enter the following command at any level of the CLI:

```
FastIron#show ip bgp config
Current BGP configuration:
router bgp
 address-filter  1 deny  any   any
 as-path-filter  1 permit ^65001$
 local-as 65002
 maximum-paths 4
 neighbor pg1 peer-group
 neighbor pg1 remote-as 65001
 neighbor pg1 description "FastIron group 1"
 neighbor pg1 distribute-list out 1
 neighbor 192.169.100.1 peer-group pg1
 neighbor 192.169.101.1 peer-group pg1
 neighbor 192.169.102.1 peer-group pg1
 neighbor 192.169.201.1 remote-as 65101
 neighbor 192.169.201.1 shutdown
 neighbor 192.169.220.3 remote-as 65432
 network 1.1.1.0 255.255.255.0
 network 2.2.2.0 255.255.255.0
 redistribute connected
```

Syntax: show ip bgp config

Displaying CPU Utilization Statistics

You can display CPU utilization statistics for BGP4 and other IP protocols.

To display CPU utilization statistics for BGP4 for the previous one-second, one-minute, five-minute, and fifteen-minute intervals, enter the following command at any level of the CLI:

```
FastIron#show process cpu
Process Name   5Sec(%)   1Min(%)   5Min(%)   15Min(%)   Runtime(ms)
ARP              0.01      0.03      0.09      0.22              9
BGP              0.04      0.06      0.08      0.14             13
GVRP             0.00      0.00      0.00      0.00              0
ICMP             0.00      0.00      0.00      0.00              0
IP               0.00      0.00      0.00      0.00              0
OSPF             0.00      0.00      0.00      0.00              0
RIP              0.00      0.00      0.00      0.00              0
STP              0.00      0.00      0.00      0.00              0
VRRP             0.00      0.00      0.00      0.00              0
```

If the software has been running less than 15 minutes (the maximum interval for utilization statistics), the command indicates how long the software has been running. Here is an example:

```
FastIron#show process cpu
The system has only been up for 6 seconds.
Process Name   5Sec(%)   1Min(%)   5Min(%)   15Min(%)   Runtime(ms)
ARP              0.01      0.00      0.00      0.00              0
BGP              0.00      0.00      0.00      0.00              0
GVRP             0.00      0.00      0.00      0.00              0
ICMP             0.01      0.00      0.00      0.00              1
IP               0.00      0.00      0.00      0.00              0
OSPF             0.00      0.00      0.00      0.00              0
RIP              0.00      0.00      0.00      0.00              0
STP              0.00      0.00      0.00      0.00              0
VRRP             0.00      0.00      0.00      0.00              0
```

To display utilization statistics for a specific number of seconds, enter a command such as the following:

```
FastIron#show process cpu 2
Statistics for last 1 sec and 80 ms
Process Name   Sec(%)   Time(ms)
ARP              0.00       0
BGP              0.00       0
GVRP             0.00       0
ICMP             0.01       1
IP               0.00       0
OSPF             0.00       0
RIP              0.00       0
STP              0.01       0
VRRP             0.00       0
```

When you specify how many seconds' worth of statistics you want to display, the software selects the sample that most closely matches the number of seconds you specified. In this example, statistics are requested for the previous two seconds. The closest sample available is actually for the previous 1 second plus 80 milliseconds.

Syntax: show process cpu [<num>]

The <num> parameter specifies the number of seconds and can be from 1 – 900. If you use this parameter, the command lists the usage statistics only for the specified number of seconds. If you do not use this parameter, the command lists the usage statistics for the previous one-second, one-minute, five-minute, and fifteen-minute intervals.

Displaying Summary Neighbor Information

To display summary neighbor information, enter a command such as the following at any level of the CLI:

```
FastIron#show ip bgp neighbor 192.168.4.211 routes-summary
1   IP Address: 192.168.4.211
Routes Accepted/Installed:1,  Filtered/Kept:11,  Filtered:11
   Routes Selected as BEST Routes:1
      BEST Routes not Installed in IP Forwarding Table:0
   Unreachable Routes (no IGP Route for NEXTHOP):0
   History Routes:0

NLRIs Received in Update Message:24,  Withdraws:0 (0),  Replacements:1
   NLRIs Discarded due to
      Maximum Prefix Limit:0, AS Loop:0
      Invalid Nexthop:0, Invalid Nexthop Address:0.0.0.0
      Duplicated Originator_ID:0, Cluster_ID:0

Routes Advertised:0,  To be Sent:0,  To be Withdrawn:0
NLRIs Sent in Update Message:0,  Withdraws:0,  Replacements:0

Peer Out of Memory Count for:
   Receiving Update Messages:0, Accepting Routes(NLRI):0
   Attributes:0, Outbound Routes(RIB-out):0
```

Syntax: show ip bgp neighbors [<ip-addr>] | [route-summary]

This display shows the following information.

Table 38.5: BGP4 Route Summary Information for a Neighbor

This Field...	Displays...
IP Address	The IP address of the neighbor
Routes Received	How many routes the Layer 3 Switch has received from the neighbor during the current BGP4 session. • Accepted/Installed – Indicates how many of the received routes the Layer 3 Switch accepted and installed in the BGP4 route table. • Filtered/Kept – Indicates how many routes were filtered out, but were nonetheless retained in memory for use by the soft reconfiguration feature. • Filtered – Indicates how many of the received routes were filtered out.
Routes Selected as BEST Routes	The number of routes that the Layer 3 Switch selected as the best routes to their destinations.
BEST Routes not Installed in IP Forwarding Table	The number of routes received from the neighbor that are the best BGP4 routes to their destinations, but were nonetheless not installed in the IP route table because the Layer 3 Switch received better routes from other sources (such as OSPF, RIP, or static IP routes).
Unreachable Routes	The number of routes received from the neighbor that are unreachable because the Layer 3 Switch does not have a valid RIP, OSPF, or static route to the next hop.

Table 38.5: BGP4 Route Summary Information for a Neighbor (Continued)

This Field...	Displays...
History Routes	The number of routes that are down but are being retained for route flap dampening purposes.
NLRIs Received in Update Message	The number of routes received in Network Layer Reachability (NLRI) format in UPDATE messages. • Withdraws – The number of withdrawn routes the Layer 3 Switch has received. • Replacements – The number of replacement routes the Layer 3 Switch has received.
NLRIs Discarded due to	Indicates the number of times the Layer 3 Switch discarded an NLRI for the neighbor due to the following reasons: • Maximum Prefix Limit – The Layer 3 Switch's configured maximum prefix amount had been reached. • AS Loop – An AS loop occurred. An AS loop occurs when the BGP4 AS-path attribute contains the local AS number. • Invalid Nexthop – The next hop value was not acceptable. • Duplicated Originator_ID – The originator ID was the same as the local router ID. • Cluster_ID – The cluster list contained the local cluster ID, or contained the local router ID (see above) if the cluster ID is not configured.
Routes Advertised	The number of routes the Layer 3 Switch has advertised to this neighbor. • To be Sent – The number of routes the Layer 3 Switch has queued to send to this neighbor. • To be Withdrawn – The number of NLRIs for withdrawing routes the Layer 3 Switch has queued up to send to this neighbor in UPDATE messages.
NLRIs Sent in Update Message	The number of NLRIs for new routes the Layer 3 Switch has sent to this neighbor in UPDATE messages. • Withdraws – The number of routes the Layer 3 Switch has sent to the neighbor to withdraw. • Replacements – The number of routes the Layer 3 Switch has sent to the neighbor to replace routes the neighbor already has.

Table 38.5: BGP4 Route Summary Information for a Neighbor (Continued)

This Field...	Displays...
Peer Out of Memory Count for	Statistics for the times the Layer 3 Switch has run out of BGP4 memory for the neighbor during the current BGP4 session. • Receiving Update Messages – The number of times UPDATE messages were discarded because there was no memory for attribute entries. • Accepting Routes(NLRI) – The number of NLRIs discarded because there was no memory for NLRI entries. This count is not included in the Receiving Update Messages count. • Attributes – The number of times there was no memory for BGP4 attribute entries. • Outbound Routes(RIB-out) – The number of times there was no memory to place a “best” route into the neighbor's route information base (Adj-RIB-Out) for routes to be advertised.

Displaying BGP4 Neighbor Information

To view BGP4 neighbor information including the values for all the configured parameters, enter the following command.

NOTE: The display shows all the configured parameters for the neighbor. Only the parameters that have values different from their defaults are shown.

```
FastIron#show ip bgp neighbor 10.4.0.2
1   IP Address: 10.4.0.2, AS: 5 (EBGP), RouterID: 100.0.0.1
       Description: neighbor 10.4.0.2
    State: ESTABLISHED, Time: 0h1m0s, KeepAliveTime: 0, HoldTime: 0
       PeerGroup: pg1
       Multihop-EBGP: yes, ttl: 1
       RouteReflectorClient: yes
       SendCommunity: yes
       NextHopSelf: yes
       DefaultOriginate: yes (default sent)
       MaximumPrefixLimit: 90000
       RemovePrivateAs: : yes
       RefreshCapability: Received
    Route Filter Policies:
       Distribute-list: (out) 20
       Filter-list: (in) 30
       Prefix-list: (in) pf1
       Route-map: (in) setnp1  (out) setnp2
    Messages:    Open    Update  KeepAlive Notification Refresh-Req
       Sent    : 1       1       1         0            0
       Received: 1       8       1         0            0
    Last Update Time: NLRI       Withdraw          NLRI       Withdraw
                  Tx: 0h0m59s    ---           Rx: 0h0m59s    ---
    Last Connection Reset Reason:Unknown
      Notification Sent:     Unspecified
      Notification Received: Unspecified
    TCP Connection state: ESTABLISHED
       Local host:  10.4.0.1, Local  Port: 179
       Remote host: 10.4.0.2, Remote Port: 8053
       ISentSeq:   52837276  SendNext:   52837392  TotUnAck:          0
       TotSent:         116  ReTrans:           0  UnAckSeq:   52837392
       IRcvSeq:  2155052043  RcvNext:  2155052536  SendWnd:       16384
       TotalRcv:        493  DupliRcv:          0  RcvWnd:        16384
       SendQue:           0  RcvQue:            0  CngstWnd:       1460
```

This example shows how to display information for a specific neighbor, by specifying the neighbor's IP address with the command. None of the other display options are used; thus, all of the information is displayed for the neighbor. The number in the far left column indicates the neighbor for which information is displayed. When you list information for multiple neighbors, this number makes the display easier to read.

The TCP statistics at the end of the display show status for the TCP session with the neighbor. Most of the fields show information stored in the Layer 3 Switch's Transmission Control Block (TCB) for the TCP session between the Layer 3 Switch and its neighbor. These fields are described in detail in section 3.2 of RFC 793, "Transmission Control Protocol Functional Specification".

Syntax: show ip bgp neighbors [<ip-addr> [advertised-routes [detail [<ip-addr>[/<mask-bits>]]]] | [attribute-entries [detail]] | [flap-statistics] | [last-packet-with-error] | [received prefix-filter] | [received-routes] | [routes [best] | [detail [best] | [not-installed-best] | [unreachable]] | [rib-out-routes [<ip-addr>/<mask-bits> | <ip-addr> <net-mask> | detail]] | [routes-summary]]

The <ip-addr> option lets you narrow the scope of the command to a specific neighbor.

The **advertised-routes** option displays only the routes that the Layer 3 Switch has advertised to the neighbor during the current BGP4 neighbor session.

The **attribute-entries** option shows the attribute-entries associated with routes received from the neighbor.

The **flap-statistics** option shows the route flap statistics for routes received from or sent to the neighbor.

The **last-packet-with-error** option displays the last packet from the neighbor that contained an error. The packet's contents are displayed in decoded (human-readable) format.

The **received prefix-filter** option shows the Outbound Route Filters (ORFs) received from the neighbor. This option applies to cooperative route filtering.

The **received-routes** option lists all the route information received in route updates from the neighbor since the soft reconfiguration feature was enabled. See “Using Soft Reconfiguration” on page 38-102.

The **routes** option lists the routes received in UPDATE messages from the neighbor. You can specify the following additional options:

- **best** – Displays the routes received from the neighbor that the Layer 3 Switch selected as the best routes to their destinations.
- **not-installed-best** – Displays the routes received from the neighbor that are the best BGP4 routes to their destinations, but were nonetheless not installed in the IP route table because the Layer 3 Switch received better routes from other sources (such as OSPF, RIP, or static IP routes).
- **unreachable** – Displays the routes that are unreachable because the Layer 3 Switch does not have a valid RIP, OSPF, or static route to the next hop.
- **detail** – Displays detailed information for the specified routes. You can refine your information request by also specifying one of the options above (**best**, **not-installed-best**, or **unreachable**).

The **rib-out-routes** option lists the route information base (RIB) for outbound routes. You can display all the routes or specify a network address.

The **routes-summary** option displays a summary of the following information:

- Number of routes received from the neighbor
- Number of routes accepted by this Layer 3 Switch from the neighbor
- Number of routes this Layer 3 Switch filtered out of the UPDATES received from the neighbor and did not accept
- Number of routes advertised to the neighbor
- Number of attribute entries associated with routes received from or advertised to the neighbor.

This display shows the following information.

Table 38.6: BGP4 Neighbor Information

This Field...	Displays...
IP Address	The IP address of the neighbor.
AS	The AS the neighbor is in.

Table 38.6: BGP4 Neighbor Information (Continued)

This Field...	Displays...
EBGP/IBGP	Whether the neighbor session is an IBGP session, an EBGP session, or a confederation EBGP session. • EBGP – The neighbor is in another AS. • EBGP_Confed – The neighbor is a member of another sub-AS in the same confederation. • IBGP – The neighbor is in the same AS.
RouterID	The neighbor’s router ID.
Description	The description you gave the neighbor when you configured it on the Layer 3 Switch.

Table 38.6: BGP4 Neighbor Information (Continued)

This Field...	Displays...
State	The state of the router's session with the neighbor. The states are from this router's perspective of the session, not the neighbor's perspective. The state values are based on the BGP4 state machine values described in RFC 1771 and can be one of the following for each router: • IDLE – The BGP4 process is waiting to be started. Usually, enabling BGP4 or establishing a neighbor session starts the BGP4 process. • A minus sign (-) indicates that the session has gone down and the software is clearing or removing routes. • ADMND – The neighbor has been administratively shut down. See "Administratively Shutting Down a Session with a BGP4 Neighbor" on page 38-19. • A minus sign (-) indicates that the session has gone down and the software is clearing or removing routes. • CONNECT – BGP4 is waiting for the connection process for the TCP neighbor session to be completed. • ACTIVE – BGP4 is waiting for a TCP connection from the neighbor. **Note**: If the state frequently changes between CONNECT and ACTIVE, there may be a problem with the TCP connection. • OPEN SENT – BGP4 is waiting for an Open message from the neighbor. • OPEN CONFIRM – BGP4 has received an OPEN message from the neighbor and is now waiting for either a KEEPALIVE or NOTIFICATION message. If the router receives a KEEPALIVE message from the neighbor, the state changes to Established. If the message is a NOTIFICATION, the state changes to Idle. • ESTABLISHED – BGP4 is ready to exchange UPDATE messages with the neighbor. • If there is more BGP data in the TCP receiver queue, a plus sign (+) is also displayed. **Note**: If you display information for the neighbor using the **show ip bgp neighbor** <ip-addr> command, the TCP receiver queue value will be greater than 0.
Time	The amount of time this session has been in its current state.
KeepAliveTime	The keep alive time, which specifies how often this router sends keep alive messages to the neighbor. See "Changing the Keep Alive Time and Hold Time" on page 38-20.
HoldTime	The hold time, which specifies how many seconds the router will wait for a KEEPALIVE or UPDATE message from a BGP4 neighbor before deciding that the neighbor is dead. See "Changing the Keep Alive Time and Hold Time" on page 38-20.
PeerGroup	The name of the peer group the neighbor is in, if applicable.

Table 38.6: BGP4 Neighbor Information (Continued)

This Field...	Displays...
Multihop-EBGP	Whether this option is enabled for the neighbor.
RouteReflectorClient	Whether this option is enabled for the neighbor.
SendCommunity	Whether this option is enabled for the neighbor.
NextHopSelf	Whether this option is enabled for the neighbor.
DefaultOriginate	Whether this option is enabled for the neighbor.
MaximumPrefixLimit	Lists the maximum number of prefixes the Layer 3 Switch will accept from this neighbor.
RemovePrivateAs	Whether this option is enabled for the neighbor.
RefreshCapability	Whether this Layer 3 Switch has received confirmation from the neighbor that the neighbor supports the dynamic refresh capability.
CooperativeFilteringCapability	Whether the neighbor is enabled for cooperative route filtering.
Distribute-list	Lists the distribute list parameters, if configured.
Filter-list	Lists the filter list parameters, if configured.
Prefix-list	Lists the prefix list parameters, if configured.
Route-map	Lists the route map parameters, if configured.
Messages Sent	The number of messages this router has sent to the neighbor. The display shows statistics for the following message types: • Open • Update • KeepAlive • Notification • Refresh-Req
Messages Received	The number of messages this router has received from the neighbor. The message types are the same as for the Message Sent field.
Last Update Time	Lists the last time updates were sent and received for the following: • NLRIs • Withdraws

Table 38.6: BGP4 Neighbor Information (Continued)

This Field...	Displays...
Last Connection Reset Reason	The reason the previous session with this neighbor ended. The reason can be one of the following: • Reasons described in the BGP specifications: • Message Header Error • Connection Not Synchronized • Bad Message Length • Bad Message Type • OPEN Message Error • Unsupported Version Number • Bad Peer AS Number • Bad BGP Identifier • Unsupported Optional Parameter • Authentication Failure • Unacceptable Hold Time • Unsupported Capability • UPDATE Message Error • Malformed Attribute List • Unrecognized Well-known Attribute • Missing Well-known Attribute • Attribute Flags Error • Attribute Length Error • Invalid ORIGIN Attribute • Invalid NEXT_HOP Attribute • Optional Attribute Error • Invalid Network Field • Malformed AS_PATH • Hold Timer Expired • Finite State Machine Error • Rcv Notification

Table 38.6: BGP4 Neighbor Information (Continued)

This Field...	Displays...
Last Connection Reset Reason (cont.)	• Reasons specific to the Foundry implementation: • Reset All Peer Sessions • User Reset Peer Session • Port State Down • Peer Removed • Peer Shutdown • Peer AS Number Change • Peer AS Confederation Change • TCP Connection KeepAlive Timeout • TCP Connection Closed by Remote • TCP Data Stream Error Detected

Table 38.6: BGP4 Neighbor Information (Continued)

This Field...	Displays...
Notification Sent	If the router receives a NOTIFICATION message from the neighbor, the message contains an error code corresponding to one of the following errors. Some errors have subcodes that clarify the reason for the error. Where applicable, the subcode messages are listed underneath the error code messages. • Message Header Error • Connection Not Synchronized • Bad Message Length • Bad Message Type • Unspecified • Open Message Error • Unsupported Version • Bad Peer As • Bad BGP Identifier • Unsupported Optional Parameter • Authentication Failure • Unacceptable Hold Time • Unspecified • Update Message Error • Malformed Attribute List • Unrecognized Attribute • Missing Attribute • Attribute Flag Error • Attribute Length Error • Invalid Origin Attribute • Invalid NextHop Attribute • Optional Attribute Error • Invalid Network Field • Malformed AS Path • Unspecified • Hold Timer Expired • Finite State Machine Error • Cease • Unspecified
Notification Received	See above.

Table 38.6: BGP4 Neighbor Information (Continued)

This Field...	Displays...
TCP Connection state	The state of the connection with the neighbor. The connection can have one of the following states: • LISTEN – Waiting for a connection request. • SYN-SENT – Waiting for a matching connection request after having sent a connection request. • SYN-RECEIVED – Waiting for a confirming connection request acknowledgment after having both received and sent a connection request. • ESTABLISHED – Data can be sent and received over the connection. This is the normal operational state of the connection. • FIN-WAIT-1 – Waiting for a connection termination request from the remote TCP, or an acknowledgment of the connection termination request previously sent. • FIN-WAIT-2 – Waiting for a connection termination request from the remote TCP. • CLOSE-WAIT – Waiting for a connection termination request from the local user. • CLOSING – Waiting for a connection termination request acknowledgment from the remote TCP. • LAST-ACK – Waiting for an acknowledgment of the connection termination request previously sent to the remote TCP (which includes an acknowledgment of its connection termination request). • TIME-WAIT – Waiting for enough time to pass to be sure the remote TCP received the acknowledgment of its connection termination request. • CLOSED – There is no connection state.
Byte Sent	The number of bytes sent.
Byte Received	The number of bytes received.
Local host	The IP address of the Layer 3 Switch.
Local port	The TCP port the Layer 3 Switch is using for the BGP4 TCP session with the neighbor.
Remote host	The IP address of the neighbor.
Remote port	The TCP port the neighbor is using for the BGP4 TCP session with the Layer 3 Switch.
ISentSeq	The initial send sequence number for the session.
SendNext	The next sequence number to be sent.
TotUnAck	The number of sequence numbers sent by the Layer 3 Switch that have not been acknowledged by the neighbor.

Table 38.6: BGP4 Neighbor Information (Continued)

This Field...	Displays...
TotSent	The number of sequence numbers sent to the neighbor.
ReTrans	The number of sequence numbers that the Layer 3 Switch retransmitted because they were not acknowledged.
UnAckSeq	The current acknowledged sequence number.
IRcvSeq	The initial receive sequence number for the session.
RcvNext	The next sequence number expected from the neighbor.
SendWnd	The size of the send window.
TotalRcv	The number of sequence numbers received from the neighbor.
DupliRcv	The number of duplicate sequence numbers received from the neighbor.
RcvWnd	The size of the receive window.
SendQue	The number of sequence numbers in the send queue.
RcvQue	The number of sequence numbers in the receive queue.
CngstWnd	The number of times the window has changed.

Displaying Route Information for a Neighbor

You can display routes based on the following criteria:

- A summary of the routes for a specific neighbor.
- The routes received from the neighbor that the Layer 3 Switch selected as the best routes to their destinations.
- The routes received from the neighbor that are the best BGP4 routes to their destinations, but were nonetheless not installed in the IP route table because the Layer 3 Switch received better routes from other sources (such as OSPF, RIP, or static IP routes).
- The routes that are unreachable because the Layer 3 Switch does not have a valid RIP, OSPF, or static route to the next hop.
- Routes for a specific network advertised by the Layer 3 Switch to the neighbor.
- The Routing Information Base (RIB) for a specific network advertised to the neighbor. You can display the RIB regardless of whether the Layer 3 Switch has already sent it to the neighbor.

To display route information for a neighbor, use the following CLI methods.

Displaying Summary Route Information

To display summary route information, enter a command such as the following at any level of the CLI:

```
FastIron#show ip bgp neighbor 10.1.0.2 routes-summary
1   IP Address: 10.1.0.2
Routes Accepted/Installed:1,  Filtered/Kept:11,  Filtered:11
   Routes Selected as BEST Routes:1
      BEST Routes not Installed in IP Forwarding Table:0
   Unreachable Routes (no IGP Route for NEXTHOP):0
   History Routes:0

NLRIs Received in Update Message:24,  Withdraws:0 (0),  Replacements:1
   NLRIs Discarded due to
      Maximum Prefix Limit:0, AS Loop:0
      Invalid Nexthop:0, Invalid Nexthop Address:0.0.0.0
      Duplicated Originator_ID:0, Cluster_ID:0

Routes Advertised:0,  To be Sent:0,  To be Withdrawn:0
NLRIs Sent in Update Message:0,  Withdraws:0,  Replacements:0

Peer Out of Memory Count for:
   Receiving Update Messages:0, Accepting Routes(NLRI):0
   Attributes:0, Outbound Routes(RIB-out):0
```

This display shows the following information.

Table 38.7: BGP4 Route Summary Information for a Neighbor

This Field...	Displays...
Routes Received	How many routes the Layer 3 Switch has received from the neighbor during the current BGP4 session. • Accepted/Installed – Indicates how many of the received routes the Layer 3 Switch accepted and installed in the BGP4 route table. • Filtered – Indicates how many of the received routes the Layer 3 Switch did not accept or install because they were denied by filters on the Layer 3 Switch.
Routes Selected as BEST Routes	The number of routes that the Layer 3 Switch selected as the best routes to their destinations.
BEST Routes not Installed in IP Forwarding Table	The number of routes received from the neighbor that are the best BGP4 routes to their destinations, but were nonetheless not installed in the IP route table because the Layer 3 Switch received better routes from other sources (such as OSPF, RIP, or static IP routes).
Unreachable Routes	The number of routes received from the neighbor that are unreachable because the Layer 3 Switch does not have a valid RIP, OSPF, or static route to the next hop.
History Routes	The number of routes that are down but are being retained for route flap dampening purposes.

Table 38.7: BGP4 Route Summary Information for a Neighbor (Continued)

This Field...	Displays...
NLRIs Received in Update Message	The number of routes received in Network Layer Reachability (NLRI) format in UPDATE messages. • Withdraws – The number of withdrawn routes the Layer 3 Switch has received. • Replacements – The number of replacement routes the Layer 3 Switch has received.
NLRIs Discarded due to	Indicates the number of times the Layer 3 Switch discarded an NLRI for the neighbor due to the following reasons: • Maximum Prefix Limit – The Layer 3 Switch's configured maximum prefix amount had been reached. • AS Loop – An AS loop occurred. An AS loop occurs when the BGP4 AS-path attribute contains the local AS number. • Invalid Nexthop – The next hop value was not acceptable. • Duplicated Originator_ID – The originator ID was the same as the local router ID. • Cluster_ID – The cluster list contained the local cluster ID, or contained the local router ID (see above) if the cluster ID is not configured.
Routes Advertised	The number of routes the Layer 3 Switch has advertised to this neighbor. • To be Sent – The number of routes the Layer 3 Switch has queued to send to this neighbor. • To be Withdrawn – The number of NLRIs for withdrawing routes the Layer 3 Switch has queued up to send to this neighbor in UPDATE messages.
NLRIs Sent in Update Message	The number of NLRIs for new routes the Layer 3 Switch has sent to this neighbor in UPDATE messages. • Withdraws – The number of routes the Layer 3 Switch has sent to the neighbor to withdraw. • Replacements – The number of routes the Layer 3 Switch has sent to the neighbor to replace routes the neighbor already has.

Table 38.7: BGP4 Route Summary Information for a Neighbor (Continued)

This Field...	Displays...
Peer Out of Memory Count for	Statistics for the times the Layer 3 Switch has run out of BGP4 memory for the neighbor during the current BGP4 session. • Receiving Update Messages – The number of times UPDATE messages were discarded because there was no memory for attribute entries. • Accepting Routes(NLRI) – The number of NLRIs discarded because there was no memory for NLRI entries. This count is not included in the Receiving Update Messages count. • Attributes – The number of times there was no memory for BGP4 attribute entries. • Outbound Routes(RIB-out) – The number of times there was no memory to place a “best” route into the neighbor's route information base (Adj-RIB-Out) for routes to be advertised.

Displaying Advertised Routes

To display the routes the Layer 3 Switch has advertised to a specific neighbor for a specific network, enter a command such as the following at any level of the CLI:

```
FastIron#show ip bgp neighbors 192.168.4.211 advertised-routes
       There are 2 routes advertised to neighbor 192.168.4.211
Status A:AGGREGATE B:BEST b:NOT-INSTALLED-BEST I:IBGP L:LOCAL
       Network        Next Hop        Metric     LocPrf     Weight     Status
1      102.0.0.0/24   192.168.2.102   12                    32768      BL
2      200.1.1.0/24   192.168.2.102   0                     32768    BL
```

You also can enter a specific route, as in the following example:

```
FastIron#show ip bgp neighbors 192.168.4.211 advertised 200.1.1.0/24
Status A:AGGREGATE B:BEST b:NOT-INSTALLED-BEST I:IBGP L:LOCAL
       Network        Next Hop        Metric     LocPrf     Weight     Status
1      200.1.1.0/24   192.168.2.102   0                     32768      BL
```

Syntax: show ip bgp neighbor <ip-addr> advertised-routes [<ip-addr>/<prefix>]

For information about the fields in this display, see Table 38.9 on page 38-93. The fields in this display also appear in the **show ip bgp** display.

Displaying the Best Routes

To display the routes received from a specific neighbor that are the “best” routes to their destinations, enter a command such as the following at any level of the CLI:

```
FastIron#show ip bgp neighbor 192.168.4.211 routes best
```

Syntax: show ip bgp neighbor <ip-addr> routes best

For information about the fields in this display, see Table 38.9 on page 38-93. The fields in this display also appear in the **show ip bgp** display.

Displaying the Best Routes that Were Nonetheless Not Installed in the IP Route Table

To display the BGP4 routes received from a specific neighbor that are the “best” routes to their destinations but are not installed in the Layer 3 Switch’s IP route table, enter a command such as the following at any level of the CLI:

```
FastIron#show ip bgp neighbor 192.168.4.211 routes not-installed-best
```

Each of the displayed routes is a valid path to its destination, but the Layer 3 Switch received another path from a different source (such as OSPF, RIP, or a static route) that has a lower administrative distance. The Layer 3 Switch always selects the path with the lowest administrative distance to install in the IP route table.

Syntax: show ip bgp neighbor <ip-addr> routes not-installed-best

For information about the fields in this display, see Table 38.9 on page 38-93. The fields in this display also appear in the **show ip bgp** display.

Displaying the Routes Whose Destinations Are Unreachable

To display BGP4 routes whose destinations are unreachable using any of the BGP4 paths in the BGP4 route table, enter a command such as the following at any level of the CLI:

```
FastIron#show ip bgp neighbor 192.168.4.211 routes unreachable
```

Syntax: show ip bgp neighbor <ip-addr> routes unreachable

For information about the fields in this display, see Table 38.9 on page 38-93. The fields in this display also appear in the **show ip bgp** display.

Displaying the Adj-RIB-Out for a Neighbor

To display the Layer 3 Switch’s current BGP4 Routing Information Base (Adj-RIB-Out) for a specific neighbor and a specific destination network, enter a command such as the following at any level of the CLI:

```
FastIron#show ip bgp neighbor 192.168.4.211 rib-out-routes 192.168.1.0/24
Status A:AGGREGATE B:BEST b:NOT-INSTALLED-BEST I:IBGP L:LOCAL
       Prefix             Next Hop        Metric     LocPrf     Weight Status
1      200.1.1.0/24       0.0.0.0         0          101        32768  BL
```

The Adj-RIB-Out contains the routes that the Layer 3 Switch either has most recently sent to the neighbor or is about to send to the neighbor.

Syntax: show ip bgp neighbor <ip-addr> rib-out-routes [<ip-addr>/<prefix>]

For information about the fields in this display, see Table 38.9 on page 38-93. The fields in this display also appear in the **show ip bgp** display.

Displaying Peer Group Information

You can display configuration information for peer groups.

To display peer-group information, enter a command such as the following at the Privileged EXEC level of the CLI:

```
FastIron#show ip bgp peer-group pg1
1   BGP peer-group is pg
    Description: peer group abc
       SendCommunity: yes
       NextHopSelf: yes
       DefaultOriginate: yes
    Members:
       IP Address: 192.168.10.10, AS: 65111
```

Syntax: show ip bgp peer-group [<peer-group-name>]

Only the parameters that have values different from their defaults are listed.

Displaying Summary Route Information

To display summary statistics for all the routes in the Layer 3 Switch's BGP4 route table, enter a command such as the following at any level of the CLI:

```
FastIron#show ip bgp routes summary
  Total number of BGP routes (NLRIs) Installed     : 20
  Distinct BGP destination networks                : 20
  Filtered BGP routes for soft reconfig            : 100178
  Routes originated by this router                 : 2
  Routes selected as BEST routes                   : 19
  BEST routes not installed in IP forwarding table : 1
  Unreachable routes (no IGP route for NEXTHOP)    : 1
  IBGP routes selected as best routes              : 0
  EBGP routes selected as best routes              : 17
```

Syntax: show ip bgp routes summary

This display shows the following information.

Table 38.8: BGP4 Summary Route Information

This Field...	Displays...
Total number of BGP routes (NLRIs) Installed	The number of BGP4 routes the Layer 3 Switch has installed in the BGP4 route table.
Distinct BGP destination networks	The number of destination networks the installed routes represent. The BGP4 route table can have multiple routes to the same network.
Filtered BGP routes for soft reconfig	The number of route updates received from soft-reconfigured neighbors or peer groups that have been filtered out but retained. For information about soft reconfiguration, see "Using Soft Reconfiguration" on page 38-102.
Routes originated by this router	The number of routes in the BGP4 route table that this Layer 3 Switch originated.
Routes selected as BEST routes	The number of routes in the BGP4 route table that this Layer 3 Switch has selected as the best routes to the destinations.
BEST routes not installed in IP forwarding table	The number of BGP4 routes that are the best BGP4 routes to their destinations but were not installed in the IP route table because the Layer 3 Switch received better routes from other sources (such as OSPF, RIP, or static IP routes).
Unreachable routes (no IGP route for NEXTHOP)	The number of routes in the BGP4 route table whose destinations are unreachable because the next hop is unreachable.
IBGP routes selected as best routes	The number of "best" routes in the BGP4 route table that are IBGP routes.
EBGP routes selected as best routes	The number of "best" routes in the BGP4 route table that are EBGP routes.

Displaying the BGP4 Route Table

BGP4 uses filters you define as well as the algorithm described in "How BGP4 Selects a Path for a Route" on page 38-3 to determine the preferred route to a destination. BGP4 sends only the preferred route to the router's IP table. However, if you want to view all the routes BGP4 knows about, you can display the BGP4 table using either of the following methods.

To view the BGP4 route table, enter the following command:

```
FastIron#show ip bgp routes
Total number of BGP Routes: 97371
Status A:AGGREGATE B:BEST b:NOT-INSTALLED-BEST C:CONFED_EBGP D:DAMPED
       E:EBGP H:HISTORY I:IBGP L:LOCAL M:MULTIPATH S:SUPPRESSED
       Prefix            Next Hop        Metric     LocPrf     Weight Status
1      3.0.0.0/8         192.168.4.106              100        0      BE
         AS_PATH: 65001 4355 701 80
2      4.0.0.0/8         192.168.4.106              100        0      BE
         AS_PATH: 65001 4355 1
3      4.60.212.0/22     192.168.4.106              100        0      BE
         AS_PATH: 65001 4355 701 1 189
4      6.0.0.0/8         192.168.4.106              100        0      BE
         AS_PATH: 65001 4355 3356 7170 1455
5      8.8.1.0/24        192.168.4.106   0          100        0      BE
         AS_PATH: 65001
```

Syntax: show ip bgp routes [[network] <ip-addr>] | <num> | [age <secs>] | [as-path-access-list <num>] | [best] | [cidr-only] | [community <num> | no-export | no-advertise | internet | local-as] | [community-access-list <num>] | [community-list <num> | [detail <option>] | [filter-list <num, num,...>] | [next-hop <ip-addr>] | [no-best] | [not-installed-best] | [prefix-list <string>] | [regular-expression <regular-expression>] | [route-map <map-name>] | [summary] | [unreachable]

The <ip-addr> option displays routes for a specific network. The **network** keyword is optional. You can enter the network address without entering "network" in front of it.

The <num> option specifies the table entry with which you want the display to start. For example, if you want to list entries beginning with table entry 100, specify 100.

The **age** <secs> parameter displays only the routes that have been received or updated more recently than the number of seconds you specify.

The **as-path-access-list** <num> parameter filters the display using the specified AS-path ACL.

The **best** parameter displays the routes received from the neighbor that the Layer 3 Switch selected as the best routes to their destinations.

The **cidr-only** option lists only the routes whose network masks do not match their class network length.

The **community** option lets you display routes for a specific community. You can specify **local-as**, **no-export**, **no-advertise**, **internet**, or a private community number. You can specify the community number as either two five-digit integer values of up to 1– 65535, separated by a colon (for example, 12345:6789) or a single long integer value.

The **community-access-list** <num> parameter filters the display using the specified community ACL.

The **community-list** option lets you display routes that match a specific community filter.

The **detail** option lets you display more details about the routes. You can refine your request by also specifying one of the other display options after the **detail** keyword.

The **filter-list** option displays routes that match a specific address filter list.

The **next-hop** <ip-addr> option displays the routes for a given next-hop IP address.

The **no-best** option displays the routes for which none of the routes to a given prefix were selected as the best route. The IP route table does not contain a BGP4 route for any of the routes listed by the command.

The **not-installed-best** option displays the routes received from the neighbor that are the best BGP4 routes to their destinations, but were nonetheless not installed in the IP route table because the Layer 3 Switch received better routes from other sources (such as OSPF, RIP, or static IP routes).

The **prefix-list** <string> parameter filters the display using the specified IP prefix list.

The **regular-expression** <regular-expression> option filters the display based on a regular expression. See "Using Regular Expressions" on page 38-45.

The **route-map** <map-name> parameter filters the display using the specified route map. The software displays only the routes that match the match statements in the route map. The software disregards the route map's set statements.

The **summary** option displays summary information for the routes.

The **unreachable** option displays the routes that are unreachable because the Layer 3 Switch does not have a valid RIP, OSPF, or static route to the next hop.

Displaying the Best BGP4 Routes

To display all the BGP4 routes in the Layer 3 Switch's BGP4 route table that are the best routes to their destinations, enter a command such as the following at any level of the CLI:

```
FastIron#show ip bgp routes best
Searching for matching routes, use ^C to quit...
Status A:AGGREGATE B:BEST b:NOT-INSTALLED-BEST C:CONFED_EBGP D:DAMPED
       E:EBGP H:HISTORY I:IBGP L:LOCAL M:MULTIPATH S:SUPPRESSED F:FILTERED
       Prefix             Next Hop        Metric     LocPrf     Weight Status
1      3.0.0.0/8          192.168.4.106              100        0      BE
         AS_PATH: 65001 4355 701 80
2      4.0.0.0/8          192.168.4.106              100        0      BE
         AS_PATH: 65001 4355 1
3      4.60.212.0/22      192.168.4.106              100        0      BE
         AS_PATH: 65001 4355 701 1 189
4      6.0.0.0/8          192.168.4.106              100        0      BE
         AS_PATH: 65001 4355 3356 7170 1455
5      9.2.0.0/16         192.168.4.106              100        0      BE
         AS_PATH: 65001 4355 701
```

Syntax: show ip bgp routes best

For information about the fields in this display, see Table 38.9 on page 38-93. The fields in this display also appear in the **show ip bgp** display.

Displaying Those Best BGP4 Routes that Are Nonetheless Not in the IP Route Table

When the Layer 3 Switch has multiple routes to a destination from different sources (such as BGP4, OSPF, RIP, or static routes), the Layer 3 Switch selects the route with the lowest administrative distance as the best route, and installs that route in the IP route table.

To display the BGP4 routes are the “best” routes to their destinations but are not installed in the Layer 3 Switch’s IP route table, enter a command such as the following at any level of the CLI:

```
FastIron#show ip bgp routes not-installed-best
Searching for matching routes, use ^C to quit...
Status A:AGGREGATE B:BEST b:NOT-INSTALLED-BEST C:CONFED_EBGP D:DAMPED
       E:EBGP H:HISTORY I:IBGP L:LOCAL M:MULTIPATH S:SUPPRESSED F:FILTERED
       Prefix             Next Hop         Metric     LocPrf     Weight Status
1      192.168.4.0/24     192.168.4.106    0          100        0      bE
         AS_PATH: 65001
```

Each of the displayed routes is a valid path to its destination, but the Layer 3 Switch received another path from a different source (such as OSPF, RIP, or a static route) that has a lower administrative distance. The Layer 3 Switch always selects the path with the lowest administrative distance to install in the IP route table.

Notice that the route status in this example is the new status, “b”. See Table 38.9 on page 38-93 for a description.

Syntax: show ip bgp routes not-installed-best

For information about the fields in this display, see Table 38.9 on page 38-93. The fields in this display also appear in the **show ip bgp** display.

NOTE: To display the routes that the Layer 3 Switch has selected as the best routes and installed in the IP route table, display the IP route table using the **show ip route** command.

Displaying BGP4 Routes Whose Destinations Are Unreachable

To display BGP4 routes whose destinations are unreachable using any of the BGP4 paths in the BGP4 route table, enter a command such as the following at any level of the CLI:

```
FastIron#show ip bgp routes unreachable
Searching for matching routes, use ^C to quit...
Status A:AGGREGATE B:BEST b:NOT-INSTALLED-BEST C:CONFED_EBGP D:DAMPED
       H:HISTORY I:IBGP L:LOCAL M:MULTIPATH S:SUPPRESSED
       Prefix      Next Hop         Metric     LocPrf     Weight Status
1      8.8.8.0/24  192.168.5.1      0          101        0
         AS_PATH: 65001 4355 1
```

Syntax: show ip bgp routes unreachable

For information about the fields in this display, see Table 38.9 on page 38-93. The fields in this display also appear in the **show ip bgp** display.

Displaying Information for a Specific Route

To display BGP4 network information by specifying an IP address within the network, enter a command such as the following at any level of the CLI:

```
FastIron#show ip bgp 9.3.4.0
Number of BGP Routes matching display condition : 1
Status codes: s suppressed, d damped, h history, * valid, > best, i internal
Origin codes: i - IGP, e - EGP, ? - incomplete
    Network            Next Hop        Metric LocPrf Weight Path
*>  9.3.4.0/24         192.168.4.106          100    0      65001 4355 1 1221 ?
       Last update to IP routing table: 0h11m38s, 1 path(s) installed:
         Gateway           Port
         192.168.2.1       2/1
       Route is advertised to 1 peers:
        20.20.20.2(65300)
```

Syntax: show ip bgp [route] <ip-addr>/<prefix> [longer-prefixes] | <ip-addr>

If you use the **route** option, the display for the information is different, as shown in the following example:

```
FastIron#show ip bgp route 9.3.4.0
Number of BGP Routes matching display condition : 1
Status A:AGGREGATE B:BEST b:NOT-INSTALLED-BEST C:CONFED_EBGP D:DAMPED
       E:EBGP H:HISTORY I:IBGP L:LOCAL M:MULTIPATH S:SUPPRESSED F:FILTERED
       Prefix             Next Hop        Metric     LocPrf     Weight Status
1      9.3.4.0/24         192.168.4.106              100        0      BE
         AS_PATH: 65001 4355 1 1221
       Last update to IP routing table: 0h12m1s, 1 path(s) installed:
         Gateway           Port
         192.168.2.1       2/1
       Route is advertised to 1 peers:
        20.20.20.2(65300)
```

These displays show the following information.

Table 38.9: BGP4 Network Information

This Field...	Displays...
Number of BGP Routes matching display condition	The number of routes that matched the display parameters you entered. This is the number of routes displayed by the command.
Status codes	A list of the characters the display uses to indicate the route's status. The status code appears in the left column of the display, to the left of each route. The status codes are described in the command's output. **Note**: This field appears only if you *do not* enter the **route** option.
Prefix	The network address and prefix.
Next Hop	The next-hop router for reaching the network from the Layer 3 Switch.
Metric	The value of the route's MED attribute. If the route does not have a metric, this field is blank.

Table 38.9: BGP4 Network Information (Continued)

This Field...	Displays...
LocPrf	The degree of preference for this route relative to other routes in the local AS. When the BGP4 algorithm compares routes on the basis of local preferences, the route with the higher local preference is chosen. The preference can have a value from 0 – 4294967295.
Weight	The value that this router associates with routes from a specific neighbor. For example, if the router receives routes to the same destination from two BGP4 neighbors, the router prefers the route from the neighbor with the larger weight.
Path	The route's AS path. **Note**: This field appears only if you *do not* enter the **route** option.
Origin code	A character the display uses to indicate the route's origin. The origin code appears to the right of the AS path (Path field). The origin codes are described in the command's output. **Note**: This field appears only if you *do not* enter the **route** option.

Table 38.9: BGP4 Network Information (Continued)

This Field...	Displays...
Status	The route's status, which can be one or more of the following: • A – AGGREGATE. The route is an aggregate route for multiple networks. • B – BEST. BGP4 has determined that this is the optimal route to the destination. **Note**: If the "b" is shown in lowercase, the software was not able to install the route in the IP route table. • b – NOT-INSTALLED-BEST. The routes received from the neighbor are the best BGP4 routes to their destinations, but were nonetheless not installed in the IP route table because the Layer 3 Switch received better routes from other sources (such as OSPF, RIP, or static IP routes). • C – CONFED_EBGP. The route was learned from a neighbor in the same confederation and AS, but in a different sub-AS within the confederation. • D – DAMPED. This route has been dampened (by the route dampening feature), and is currently unusable. • H – HISTORY. Route dampening is configured for this route, and the route has a history of flapping and is unreachable now. • I – INTERNAL. The route was learned through BGP4. • L – LOCAL. The route originated on this Layer 3 Switch. • M – MULTIPATH. BGP4 load sharing is enabled and this route was selected as one of the best ones to the destination. The best route among the multiple paths also is marked with "B". **Note**: If the "m" is shown in lowercase, the software was not able to install the route in the IP route table. • S – SUPPRESSED. This route was suppressed during aggregation and thus is not advertised to neighbors. **Note**: This field appears only if you enter the **route** option.

Displaying Route Details

Here is an example of the information displayed when you use the **detail** option. In this example, the information for one route is shown.

```
FastIron#show ip bgp routes detail
Total number of BGP Routes: 2
Status A:AGGREGATE B:BEST b:NOT-INSTALLED-BEST C:CONFED_EBGP D:DAMPED
       E:EBGP H:HISTORY I:IBGP L:LOCAL M:MULTIPATH S:SUPPRESSED
1       Prefix: 10.5.0.0/24,  Status: BME,  Age: 0h28m28s
         NEXT_HOP: 201.1.1.2,  Learned from Peer: 10.1.0.2 (5)
          LOCAL_PREF: 101,  MED: 0,  ORIGIN: igp,  Weight: 10
           AS_PATH: 5
            Adj_RIB_out count: 4,  Admin distance 20
```

These displays show the following information.

Table 38.10: BGP4 Route Information

This Field...	Displays...
Total number of BGP Routes	The number of BGP4 routes.
Status codes	A list of the characters the display uses to indicate the route's status. The status code is appears in the left column of the display, to the left of each route. The status codes are described in the command's output.
Prefix	The network prefix and mask length.
Status	The route's status, which can be one or more of the following: • A – AGGREGATE. The route is an aggregate route for multiple networks. • B – BEST. BGP4 has determined that this is the optimal route to the destination. **Note**: If the "b" is shown in lowercase, the software was not able to install the route in the IP route table. • b – NOT-INSTALLED-BEST. The routes received from the neighbor are the best BGP4 routes to their destinations, but were nonetheless not installed in the IP route table because the Layer 3 Switch received better routes from other sources (such as OSPF, RIP, or static IP routes). • C – CONFED_EBGP. The route was learned from a neighbor in the same confederation and AS, but in a different sub-AS within the confederation. • D – DAMPED. This route has been dampened (by the route dampening feature), and is currently unusable. • H – HISTORY. Route dampening is configured for this route, and the route has a history of flapping and is unreachable now. • I – INTERNAL. The route was learned through BGP4. • L – LOCAL. The route originated on this Layer 3 Switch. • M – MULTIPATH. BGP4 load sharing is enabled and this route was selected as one of the best ones to the destination. The best route among the multiple paths also is marked with "B". **Note**: If the "m" is shown in lowercase, the software was not able to install the route in the IP route table. • S – SUPPRESSED. This route was suppressed during aggregation and thus is not advertised to neighbors.
Age	The last time an update occurred.
Next_Hop	The next-hop router for reaching the network from the Layer 3 Switch.
Learned from Peer	The IP address of the neighbor that sent this route.

Table 38.10: BGP4 Route Information (Continued)

This Field...	**Displays...**
Local_Pref	The degree of preference for this route relative to other routes in the local AS. When the BGP4 algorithm compares routes on the basis of local preferences, the route with the higher local preference is chosen. The preference can have a value from 0 – 4294967295.
MED	The route's metric. If the route does not have a metric, this field is blank.
Origin	The source of the route information. The origin can be one of the following: • EGP – The routes with this set of attributes came to BGP through EGP. • IGP – The routes with this set of attributes came to BGP through IGP. • INCOMPLETE – The routes came from an origin other than one of the above. For example, they may have been redistributed from OSPF or RIP. When BGP4 compares multiple routes to a destination to select the best route, IGP is preferred over EGP and both are preferred over INCOMPLETE.
Weight	The value that this router associates with routes from a specific neighbor. For example, if the router receives routes to the same destination from two BGP4 neighbors, the router prefers the route from the neighbor with the larger weight.
Atomic	Whether network information in this route has been aggregated *and* this aggregation has resulted in information loss. **Note**: Information loss under these circumstances is a normal part of BGP4 and does not indicate an error.
Aggregation ID	The router that originated this aggregator.
Aggregation AS	The AS in which the network information was aggregated. This value applies only to aggregated routes and is otherwise 0.
Originator	The originator of the route in a route reflector environment.
Cluster List	The route-reflector clusters through which this route has passed.
Learned From	The IP address of the neighbor from which the Layer 3 Switch learned the route.
Admin Distance	The administrative distance of the route.
Adj_RIB_out	The number of neighbors to which the route has been or will be advertised. This is the number of times the route has been selected as the best route and placed in the Adj-RIB-Out (outbound queue) for a BGP4 neighbor.
Communities	The communities the route is in.

Displaying BGP4 Route-Attribute Entries

The route-attribute entries table lists the sets of BGP4 attributes stored in the router's memory. Each set of attributes is unique and can be associated with one or more routes. In fact, the router typically has fewer route attribute entries than routes. To display the route-attribute entries table, use one of the following methods.

To display the IP route table, enter the following command:

```
FastIron#show ip bgp attribute-entries
```

Syntax: show ip bgp attribute-entries

Here is an example of the information displayed by this command. A zero value indicates that the attribute is not set.

```
FastIron#show ip bgp attribute-entries
        Total number of BGP Attribute Entries: 7753
1       Next Hop  :192.168.11.1       Metric   :0                Origin:IGP
        Originator:0.0.0.0            Cluster List:None
         Aggregator:AS Number :0         Router-ID:0.0.0.0          Atomic:FALSE
        Local Pref:100                Communities:Internet
        AS Path   :(65002) 65001 4355 2548 3561 5400 6669 5548
2       Next Hop  :192.168.11.1       Metric   :0                Origin:IGP
        Originator:0.0.0.0            Cluster List:None
         Aggregator:AS Number :0         Router-ID:0.0.0.0          Atomic:FALSE
        Local Pref:100                Communities:Internet
        AS Path   :(65002) 65001 4355 2548
```

This display shows the following information.

Table 38.11: BGP4 Route-Attribute Entries Information

This Field...	Displays...
Total number of BGP Attribute Entries	The number of routes contained in this router's BGP4 route table.
Next Hop	The IP address of the next hop router for routes that have this set of attributes.
Metric	The cost of the routes that have this set of attributes.
Origin	The source of the route information. The origin can be one of the following: • EGP – The routes with this set of attributes came to BGP through EGP. • IGP – The routes with this set of attributes came to BGP through IGP. • INCOMPLETE – The routes came from an origin other than one of the above. For example, they may have been redistributed from OSPF or RIP. When BGP4 compares multiple routes to a destination to select the best route, IGP is preferred over EGP and both are preferred over INCOMPLETE.
Originator	The originator of the route in a route reflector environment.

Table 38.11: BGP4 Route-Attribute Entries Information (Continued)

This Field...	Displays...
Cluster List	The route-reflector clusters through which this set of attributes has passed.
Aggregator	Aggregator information: • `AS Number` shows the AS in which the network information in the attribute set was aggregated. This value applies only to aggregated routes and is otherwise 0. • `Router-ID` shows the router that originated this aggregator.
Atomic	Whether the network information in this set of attributes has been aggregated *and* this aggregation has resulted in information loss. • TRUE – Indicates information loss has occurred • FALSE – Indicates no information loss has occurred **Note**: Information loss under these circumstances is a normal part of BGP4 and does not indicate an error.
Local Pref	The degree of preference for routes that use this set of attributes relative to other routes in the local AS.
Communities	The communities that routes with this set of attributes are in.
AS Path	The ASs through which routes with this set of attributes have passed. The local AS is shown in parentheses.

Displaying the Routes BGP4 Has Placed in the IP Route Table

The IP route table indicates the routes it has received from BGP4 by listing “BGP” as the route type.

To display the IP route table, enter the following command:

```
FastIron#show ip route
```

Syntax: show ip route [<ip-addr> | <num> | bgp | ospf | rip]

Here is an example of the information displayed by this command. Notice that most of the routes in this example have type “B”, indicating that their source is BGP4.

```
FastIron#show ip route

Total number of IP routes: 50834

B:BGP D:Directly-Connected  O:OSPF  R:RIP  S:Static

      Network Address  NetMask          Gateway          Port      Cost   Type
      3.0.0.0          255.0.0.0        192.168.13.2     1/1       0      B
      4.0.0.0          255.0.0.0        192.168.13.2     1/1       0      B
      9.20.0.0         255.255.128.0    192.168.13.2     1/1       0      B
      10.1.0.0         255.255.0.0      0.0.0.0          1/1       1      D
      10.10.11.0       255.255.255.0    0.0.0.0          2/24      1      D
      12.2.97.0        255.255.255.0    192.168.13.2     1/1       0      B
      12.3.63.0        255.255.255.0    192.168.13.2     1/1       0      B
      12.3.123.0       255.255.255.0    192.168.13.2     1/1       0      B
      12.5.252.0       255.255.254.0    192.168.13.2     1/1       0      B
      12.6.42.0        255.255.254.0    192.168.13.2     1/1       0      B
```

remaining 50824 entries not shown...

Displaying Route Flap Dampening Statistics

To display route dampening statistics or all the dampened routes, enter the following command at any level of the CLI:

```
FastIron#show ip bgp flap-statistics

Total number of flapping routes: 414
    Status Code  >:best d:damped h:history *:valid
    Network            From            Flaps Since    Reuse    Path
h>  192.50.206.0/23    166.90.213.77   1     0 :0 :13 0 :0 :0  65001 4355 1 701
h>  203.255.192.0/20   166.90.213.77   1     0 :0 :13 0 :0 :0  65001 4355 1 7018
h>  203.252.165.0/24   166.90.213.77   1     0 :0 :13 0 :0 :0  65001 4355 1 7018
h>  192.50.208.0/23    166.90.213.77   1     0 :0 :13 0 :0 :0  65001 4355 1 701
h>  133.33.0.0/16      166.90.213.77   1     0 :0 :13 0 :0 :0  65001 4355 1 701
*>  204.17.220.0/24    166.90.213.77   1     0 :1 :4  0 :0 :0  65001 4355 701 62
```

Syntax: show ip bgp flap-statistics [regular-expression <regular-expression> | <address> <mask> [longer-prefixes] | neighbor <ip-addr> | filter-list <num>...]

The **regular-expression** <regular-expression> parameter is a regular expression. The regular expressions are the same ones supported for BGP4 AS-path filters. See "Using Regular Expressions" on page 38-45.

The <address> <mask> parameter specifies a particular route. If you also use the optional **longer-prefixes** parameter, then all statistics for routes that match the specified route or have a longer prefix than the specified route are displayed. For example, if you specify **209.157.0.0 longer**, then all routes with the prefix 209.157 or that have a longer prefix (such as 209.157.22) are displayed.

The **neighbor** <ip-addr> parameter displays route flap dampening statistics only for routes learned from the specified neighbor. You also can display route flap statistics for routes learned from a neighbor by entering the following command: **show ip bgp neighbor** <ip-addr> **flap-statistics**.

The **filter-list** <num> parameter specifies one or more filters. Only the routes that have been dampened and that match the specified filter(s) are displayed.

This display shows the following information.

Table 38.12: Route Flap Dampening Statistics

This Field...	Displays...
Total number of flapping routes	The total number of routes in the Layer 3 Switch's BGP4 route table that have changed state and thus have been marked as flapping routes.
Status code	Indicates the dampening status of the route, which can be one of the following: • > – This is the best route among those in the BGP4 route table to the route's destination. • d – This route is currently dampened, and thus unusable. • h – The route has a history of flapping and is unreachable now. • * – The route has a history of flapping but is currently usable.
Network	The destination network of the route.
From	The neighbor that sent the route to the Layer 3 Switch.

Table 38.12: Route Flap Dampening Statistics

This Field...	Displays...
Flaps	The number of flaps (state changes) the route has experienced.
Since	The amount of time since the first flap of this route.
Reuse	The amount of time remaining until this route will be un-suppressed and thus be usable again.
Path	Shows the AS-path information for the route.

You also can display all the dampened routes by entering the following command: **show ip bgp dampened-paths**.

Displaying the Active Route Map Configuration

To view the device's active route map configuration (contained in the running-config) without displaying the entire running-config, enter the following command at any level of the CLI:

```
FastIron#show route-map

route-map permitnet4 permit 10
 match ip address prefix-list plist1
route-map permitnet1 permit 1
 match ip address prefix-list plist2
route-map setcomm permit 1
 set community 1234:2345 no-export
route-map test111 permit 111
 match address-filters 11
 set community 11:12 no-export
route-map permit1122 permit 12
 match ip address 11
route-map permit1122 permit 13
 match ip address std_22
```

This example shows that the running-config contains six route maps. Notice that the match and set statements within each route map are listed beneath the command for the route map itself. In this simplified example, each route map contains only one match or set statement.

To display the active configuration for a specific route map, enter a command such as the following, which specifies a route map name:

```
FastIron#show route-map setcomm
route-map setcomm permit 1
 set community 1234:2345 no-export
```

This example shows the active configuration for a route map called “setcomm“.

Syntax: show route-map [<map-name>]

Updating Route Information and Resetting a Neighbor Session

The following sections describe ways to update route information with a neighbor, reset the session with a neighbor, and close a session with a neighbor.

Whenever you change a policy (ACL, route map, and so on) that affects the routes that the Layer 3 Switch learns from a BGP4 neighbor or peer group of neighbors, you must enter a command to place the changes into effect.

The changes take place automatically, but only affect new route updates. To make changes retroactive for routes received or sent before the changes were made, you need to enter a clear command.

You can update the learned routes using either of the following methods:

- Request the complete BGP4 route table from the neighbor or peer group. You can use this method if the neighbor supports the refresh capability (RFCs 2842 and 2858).
- Clear (reset) the session with the neighbor or peer group. This is the only method you can use if the neighbor does not support the refresh capability.

Each of these methods is effective, but can be disruptive to the network. The first method adds overhead while the Layer 3 Switch learns and filters the neighbor's or group's entire route table, while the second method adds more overhead while the devices re-establish their BGP4 sessions.

You also can clear and reset the BGP4 routes that have been installed in the IP route table. See "Clearing and Resetting BGP4 Routes in the IP Route Table" on page 38-107.

Using Soft Reconfiguration

The ***soft reconfiguration*** feature places policy changes into effect without resetting the BGP4 session. Soft reconfiguration does not request the neighbor or group to send its entire BGP4 table, nor does the feature reset the session with the neighbor or group. Instead, the soft reconfiguration feature stores all the route updates received from the neighbor or group. When you request a soft reset of inbound routes, the software performs route selection by comparing the policies against the stored route updates, instead of requesting the neighbor's BGP4 route table or resetting the session with the neighbor.

When you enable the soft reconfiguration feature, it sends a refresh message to the neighbor or group if the neighbor or group supports dynamic refresh. Otherwise, the feature resets the neighbor session. This step is required to ensure that the soft reconfiguration feature has a complete set of updates to use, and occurs only once, when you enable the feature. The feature accumulates all the route updates from the neighbor, eliminating the need for additional refreshes or resets when you change policies in the future.

To use soft reconfiguration:

- Enable the feature.
- Make the policy changes.
- Apply the changes by requesting a soft reset of the inbound updates from the neighbor or group.

Use the following CLI methods to configure soft configuration, apply policy changes, and display information for the updates that are filtered out by the policies.

Enabling Soft Reconfiguration

To configure a neighbor for soft reconfiguration, enter a command such as the following:

```
FastIron(config-bgp-router)#neighbor 10.10.200.102 soft-reconfiguration inbound
```

This command enables soft reconfiguration for updates received from 10.10.200.102. The software dynamically refreshes or resets the session with the neighbor, then retains all route updates from the neighbor following the reset.

Syntax: [no] neighbor <ip-addr> | <peer-group-name> soft-reconfiguration inbound

NOTE: The syntax related to soft reconfiguration is shown. For complete command syntax, see "Adding BGP4 Neighbors" on page 38-11.

Placing a Policy Change into Effect

To place policy changes into effect, enter a command such as the following:

```
FastIron(config-bgp-router)#clear ip bgp neighbor 10.10.200.102 soft in
```

This command updates the routes by comparing the route policies against the route updates that the Layer 3 Switch has stored. The command does not request additional updates from the neighbor or otherwise affect the session with the neighbor.

Syntax: clear ip bgp neighbor <ip-addr> | <peer-group-name> soft in

NOTE: If you do not specify "in", the command applies to both inbound and outbound updates.

NOTE: The syntax related to soft reconfiguration is shown. For complete command syntax, see "Dynamically Refreshing Routes" on page 38-105.

Displaying the Filtered Routes Received from the Neighbor or Peer Group

When you enable soft reconfiguration, the Layer 3 Switch saves all updates received from the specified neighbor or peer group. This includes updates that contain routes that are filtered out by the BGP4 route policies in effect on the Layer 3 Switch. To display the routes that have been filtered out, enter the following command at any level of the CLI:

```
FastIron#show ip bgp filtered-routes
Searching for matching routes, use ^C to quit...
Status A:AGGREGATE B:BEST b:NOT-INSTALLED-BEST C:CONFED_EBGP D:DAMPED
       E:EBGP H:HISTORY I:IBGP L:LOCAL M:MULTIPATH S:SUPPRESSED F:FILTERED
       Prefix             Next Hop        Metric     LocPrf     Weight Status
1      3.0.0.0/8          192.168.4.106              100        0      EF
         AS_PATH: 65001 4355 701 80
2      4.0.0.0/8          192.168.4.106              100        0      EF
         AS_PATH: 65001 4355 1
3      4.60.212.0/22      192.168.4.106              100        0      EF
         AS_PATH: 65001 4355 701 1 189
```

The routes displayed by the command are the routes that the Layer 3 Switch's BGP4 policies filtered out. The Layer 3 Switch did not place the routes in the BGP4 route table, but did keep the updates. If a policy change causes these routes to be permitted, the Layer 3 Switch does not need to request the route information from the neighbor, but instead uses the information in the updates.

Syntax: show ip bgp filtered-routes [<ip-addr>] | [as-path-access-list <num>] | [detail] | [prefix-list <string>]

The <ip-addr> parameter specifies the IP address of the destination network.

The **as-path-access-list** <num> parameter specifies an AS-path ACL. Only the routes permitted by the AS-path ACL are displayed.

The **detail** parameter displays detailed information for the routes. (The example above shows summary information.) You can specify any of the other options after **detail** to further refine the display request.

The prefix-list <string> parameter specifies an IP prefix list. Only the routes permitted by the prefix list are displayed.

NOTE: The syntax for displaying filtered routes is shown. For complete command syntax, see "Displaying the BGP4 Route Table" on page 38-90.

Displaying All the Routes Received from the Neighbor

To display all the route information received in route updates from a neighbor since you enabled soft reconfiguration, enter a command such as the following at any level of the CLI:

```
FastIron#show ip bgp neighbor 192.168.4.106 received-routes
       There are 97345 received routes from  neighbor 192.168.4.106
Searching for matching routes, use ^C to quit...
Status A:AGGREGATE B:BEST b:NOT-INSTALLED-BEST C:CONFED_EBGP D:DAMPED
       E:EBGP H:HISTORY I:IBGP L:LOCAL M:MULTIPATH S:SUPPRESSED F:FILTERED
       Prefix             Next Hop        Metric     LocPrf     Weight Status
1      3.0.0.0/8          192.168.4.106              100        0      BE
         AS_PATH: 65001 4355 701 80
2      4.0.0.0/8          192.168.4.106              100        0      BE
         AS_PATH: 65001 4355 1
3      4.60.212.0/22      192.168.4.106              100        0      BE
         AS_PATH: 65001 4355 701 1 189
4      6.0.0.0/8          192.168.4.106              100        0      BE
```

Syntax: show ip bgp neighbors <ip-addr> received-routes [detail]

The **detail** parameter displays detailed information for the routes. The example above shows summary information.

NOTE: The syntax for displaying received routes is shown. For complete command syntax, see "Displaying BGP4 Neighbor Information" on page 38-75.

NOTE: The **show ip bgp neighbor** <ip-addr> **received-routes** syntax supported in previous software releases is changed to the following syntax: **show ip bgp neighbor** <ip-addr> **routes**.

Dynamically Requesting a Route Refresh from a BGP4 Neighbor

You can easily apply changes to filters that control BGP4 routes received from or advertised to a neighbor, without resetting the BGP4 session between the Layer 3 Switch and the neighbor. For example, if you add, change, or remove a BGP4 address filter that denies specific routes received from a neighbor, you can apply the filter change by requesting a route refresh from the neighbor. If the neighbor also supports dynamic route refreshes, the neighbor resends its Adj-RIB-Out, its table of BGP4 routes. Using the route refresh feature, you do not need to reset the session with the neighbor.

The route refresh feature is based on the following specifications:

- RFC 2842. This RFC specifies the Capability Advertisement, which a BGP4 router uses to dynamically negotiate a capability with a neighbor.
- RFC 2858 for Multi-protocol Extension.

  NOTE: The Foundry implementation of dynamic route refresh supports negotiation of IP version 4 unicasts only.

- RFC 2918, which describes the dynamic route refresh capability

The dynamic route refresh capability is enabled by default and cannot be disabled. When the Layer 3 Switch sends a BGP4 OPEN message to a neighbor, the Layer 3 Switch includes a Capability Advertisement to inform the neighbor that the Layer 3 Switch supports dynamic route refresh.

NOTE: The option for dynamically refreshing routes received from a neighbor requires the neighbor to support dynamic route refresh. If the neighbor does not support this feature, the option does not take effect and the software displays an error message. The option for dynamically re-advertising routes to a neighbor does not require the neighbor to support dynamic route refresh.

To use the dynamic refresh feature, use either of the following methods.

Dynamically Refreshing Routes

The following sections describe how to dynamically refresh BGP4 routes to place new or changed filters into effect.

To request a dynamic refresh of all routes from a neighbor, enter a command such as the following:

```
FastIron(config-bgp-router)#clear ip bgp neighbor 192.168.1.170 soft in
```

This command asks the neighbor to send its BGP4 table (Adj-RIB-Out) again. The Layer 3 Switch applies its filters to the incoming routes and adds, modifies, or removes BGP4 routes as necessary.

Syntax: clear ip bgp neighbor all | <ip-addr> | <peer-group-name> | <as-num> [soft-outbound | soft [in | out]]

The **all** | <ip-addr> | <peer-group-name> | <as-num> specifies the neighbor. The <ip-addr> parameter specifies a neighbor by its IP interface with the Layer 3 Switch. The <peer-group-name> specifies all neighbors in a specific peer group. The <as-num> parameter specifies all neighbors within the specified AS. The **all** parameter specifies all neighbors.

The **soft-outbound** parameter updates all outbound routes by applying the new or changed filters, but sends only the existing routes affected by the new or changed filters to the neighbor.

The **soft [in | out]** parameter specifies whether you want to refresh the routes received from the neighbor or sent to the neighbor:

- **soft in** does one of the following:
  - If you enabled soft reconfiguration for the neighbor or peer group, **soft in** updates the routes by comparing the route policies against the route updates that the Layer 3 Switch has stored. Soft reconfiguration does not request additional updates from the neighbor or otherwise affect the session with the neighbor. See “Using Soft Reconfiguration” on page 38-102.
  - If you did not enable soft reconfiguration, **soft in** requests the neighbor’s entire BGP4 route table (Adj-RIB-Out), then applies the filters to add, change, or exclude routes.
  - If a neighbor does not support dynamic refresh, **soft in** resets the neighbor session.
- **soft out** updates all outbound routes, then sends the Layer 3 Switch’s entire BGP4 route table (Adj-RIB-Out) to the neighbor, after changing or excluding the routes affected by the filters.

If you do not specify **in** or **out**, the Layer 3 Switch performs both options.

NOTE: The **soft-outbound** parameter updates all outbound routes by applying the new or changed filters, but sends only the existing routes affected by the new or changed filters to the neighbor. The **soft out** parameter updates all outbound routes, then sends the Layer 3 Switch’s entire BGP4 route table (Adj-RIB-Out) to the neighbor, after changing or excluding the routes affected by the filters. Use **soft-outbound** if only the outbound policy is changed.

To dynamically resend all the Layer 3 Switch’s BGP4 routes to a neighbor, enter a command such as the following:

```
FastIron(config-bgp-router)#clear ip bgp neighbor 192.168.1.170 soft out
```

This command applies its filters for outgoing routes to the Layer 3 Switch’s BGP4 route table (Adj-RIB-Out), changes or excludes routes accordingly, then sends the resulting Adj-RIB-Out to the neighbor.

NOTE: The Foundry Layer 3 Switch does not automatically update outbound routes using a new or changed outbound policy or filter when a session with the neighbor goes up or down. Instead, the Layer 3 Switch applies a new or changed policy or filter when a route is placed in the outbound queue (Adj-RIB-Out).

To place a new or changed outbound policy or filter into effect, you must enter a **clear ip bgp neighbor** command regardless of whether the neighbor session is up or down. You can enter the command without optional parameters or with the **soft out** or **soft-outbound** option. Either way, you must specify a parameter for the neighbor (<ip-addr>, <as-num>, <peer-group-name>, or **all**).

Displaying Dynamic Refresh Information

You can use the **show ip bgp neighbors** command to display information for dynamic refresh requests. For each neighbor, the display lists the number of dynamic refresh requests the Layer 3 Switch has sent to or received from the neighbor and indicates whether the Layer 3 Switch received confirmation from the neighbor that the neighbor supports dynamic route refresh.

The RefreshCapability field indicates whether this Layer 3 Switch has received confirmation from the neighbor that the neighbor supports the dynamic refresh capability. The statistics in the Message Sent and Message Received rows under Refresh-Req indicate how many dynamic refreshes have been sent to and received from the neighbor. The statistic is cumulative across sessions.

```
FastIron#show ip bgp neighbor 10.4.0.2
1   IP Address: 10.4.0.2, AS: 5 (EBGP), RouterID: 100.0.0.1
       Description: neighbor 10.4.0.2
    State: ESTABLISHED, Time: 0h1m0s, KeepAliveTime: 0, HoldTime: 0
       PeerGroup: pg1
       Mutihop-EBGP: yes, ttl: 1
       RouteReflectorClient: yes
       SendCommunity: yes
       NextHopSelf: yes
       DefaultOriginate: yes (default sent)
       MaximumPrefixLimit: 90000
       RemovePrivateAs: : yes
       RefreshCapability: Received
    Route Filter Policies:
       Distribute-list: (out) 20
       Filter-list: (in) 30
       Prefix-list: (in) pf1
       Route-map: (in) setnp1  (out) setnp2
    Messages:    Open    Update  KeepAlive Notification Refresh-Req
       Sent    : 1       1       1         0            0
       Received: 1       8       1         0            0
    Last Update Time: NLRI       Withdraw          NLRI       Withdraw
                  Tx: 0h0m59s    ---           Rx: 0h0m59s    ---
    Last Connection Reset Reason:Unknown
      Notification Sent:     Unspecified
      Notification Received: Unspecified
    TCP Connection state: ESTABLISHED
       Byte Sent:   115, Received: 492
       Local host:  10.4.0.1, Local  Port: 179
       Remote host: 10.4.0.2, Remote Port: 8053
       ISentSeq:   52837276  SendNext:   52837392  TotUnAck:          0
       TotSent:         116  ReTrans:           0  UnAckSeq:   52837392
       IRcvSeq:  2155052043  RcvNext:  2155052536  SendWnd:       16384
       TotalRcv:        493  DupliRcv:          0  RcvWnd:        16384
       SendQue:           0  RcvQue:            0  CngstWnd:       1460
```

Closing or Resetting a Neighbor Session

You can close a neighbor session or resend route updates to a neighbor.

If you make changes to filters or route maps and the neighbor does not support dynamic route refresh, use these methods to ensure that neighbors contain only the routes you want them to contain.

- If you close a neighbor session, the Layer 3 Switch and the neighbor clear all the routes they learned from each other. When the Layer 3 Switch and neighbor establish a new BGP4 session, they exchange route tables again. Use this method if you want the Layer 3 Switch to relearn routes from the neighbor and resend its own route table to the neighbor.
- If you use the soft-outbound option, the Layer 3 Switch compiles a list of all the routes it would normally send to the neighbor at the beginning of a session. However, before sending the updates, the Foundry Layer 3 Switch also applies the filters and route maps you have configured to the list of routes. If the filters or route maps result in changes to the list of routes, the Layer 3 Switch sends updates to advertise, change, or even withdraw routes on the neighbor as needed. This ensures that the neighbor receives only the routes you want it to contain. Even if the neighbor already contains a route learned from the Layer 3 Switch that you later decided to filter out, using the soft-outbound option removes that route from the neighbor.

You can specify a single neighbor or a peer group.

To close a neighbor session and thus flush all the routes exchanged by the Layer 3 Switch and the neighbor, enter the following command:

```
FastIron#clear ip bgp neighbor all
```

Syntax: clear ip bgp neighbor all | <ip-addr> | <peer-group-name> | <as-num> [soft-outbound | soft [in | out]]

The **all** | <ip-addr> | <peer-group-name> | <as-num> specifies the neighbor. The <ip-addr> parameter specifies a neighbor by its IP interface with the Layer 3 Switch. The <peer-group-name> specifies all neighbors in a specific peer group. The <as-num> parameter specifies all neighbors within the specified AS. The **all** parameter specifies all neighbors.

To resend routes to a neighbor without closing the neighbor session, enter a command such as the following:

```
FastIron#clear ip bgp neighbor 10.0.0.1 soft out
```

Clearing and Resetting BGP4 Routes in the IP Route Table

To clear BGP4 routes from the IP route table and reset the routes, enter a command such as the following:

```
FastIron#clear ip bgp routes
```

Syntax: clear ip bgp routes [<ip-addr>/<prefix-length>]

NOTE: The **clear ip bgp routes** command has the same effect as the **clear ip route** command, but applies only to routes that come from BGP4.

Clearing Traffic Counters

You can clear the counters (reset them to 0) for BGP4 messages. To do so, use one of the following methods.

To clear the BGP4 message counter for all neighbors, enter the following command:

```
FastIron#clear ip bgp traffic
```

Syntax: clear ip bgp traffic

To clear the BGP4 message counter for a specific neighbor, enter a command such as the following:

```
FastIron#clear ip bgp neighbor 10.0.0.1 traffic
```

To clear the BGP4 message counter for all neighbors within a peer group, enter a command such as the following:

```
FastIron#clear ip bgp neighbor PeerGroup1 traffic
```

Syntax: clear ip bgp neighbor all | <ip-addr> | <peer-group-name> | <as-num> traffic

The **all** | <ip-addr> | <peer-group-name> | <as-num> specifies the neighbor. The <ip-addr> parameter specifies a neighbor by its IP interface with the Layer 3 Switch. The <peer-group-name> specifies all neighbors in a specific peer group. The <as-num> parameter specifies all neighbors within the specified AS. The **all** parameter specifies all neighbors.

Clearing Route Flap Dampening Statistics

To clear route flap dampening statistics, use the following CLI method.

NOTE: Clearing the dampening statistics for a route does not change the dampening status of the route.

To clear all the route dampening statistics, enter the following command at any level of the CLI:

```
FastIron#clear ip bgp flap-statistics
```

Syntax: clear ip bgp flap-statistics [regular-expression <regular-expression> | <address> <mask> | neighbor <ip-addr>]

The parameters are the same as those for the **show ip bgp flap-statistics** command (except the **longer-prefixes** option is not supported). See “Displaying Route Flap Dampening Statistics” on page 38-65.

NOTE: The **clear ip bgp damping** command not only clears statistics but also un-suppresses the routes. See “Displaying Route Flap Dampening Statistics” on page 38-65.

Removing Route Flap Dampening

You can un-suppress routes by removing route flap dampening from the routes. The Layer 3 Switch allows you to un-suppress all routes at once or un-suppress individual routes.

To un-suppress all the suppressed routes, enter the following command at the Privileged EXEC level of the CLI:

```
FastIron#clear ip bgp damping
```

Syntax: clear ip bgp damping [<ip-addr> <ip-mask>]

The <ip-addr> parameter specifies a particular network.

The <ip-mask> parameter specifies the network’s mask.

To un-suppress a specific route, enter a command such as the following:

```
FastIron#clear ip bgp damping 209.157.22.0 255.255.255.0
```

This command un-suppresses only the route(s) for network 209.157.22.0/24.

Clearing Diagnostic Buffers

The Layer 3 Switch stores the following BGP4 diagnostic information in buffers:

- The first 400 bytes of the last packet that contained an error
- The last NOTIFICATION message either sent or received by the Layer 3 Switch

To display these buffers, use options with the **show ip bgp neighbors** command. See “Displaying BGP4 Neighbor Information” on page 38-75.

This information can be useful if you are working with Foundry Technical Support to resolve a problem. The buffers do not identify the system time when the data was written to the buffer. If you want to ensure that diagnostic data in a buffer is recent, you can clear the buffers. You can clear the buffers for a specific neighbor or for all neighbors.

If you clear the buffer containing the first 400 bytes of the last packet that contained errors, all the bytes are changed to zeros. The Last Connection Reset Reason field of the BGP neighbor table also is cleared.

If you clear the buffer containing the last NOTIFICATION message sent or received, the buffer contains no data.

You can clear the buffers for all neighbors, for an individual neighbor, or for all the neighbors within a specific peer group.

To clear these buffers for neighbor 10.0.0.1, enter the following commands:

```
FastIron#clear ip bgp neighbor 10.0.0.1 last-packet-with-error
FastIron#clear ip bgp neighbor 10.0.0.1 notification-errors
```

Syntax: clear ip bgp neighbor all | <ip-addr> | <peer-group-name> | <as-num> last-packet-with-error | notification-errors

The **all** | <ip-addr> | <peer-group-name> | <as-num> specifies the neighbor. The <ip-addr> parameter specifies a neighbor by its IP interface with the Layer 3 Switch. The <peer-group-name> specifies all neighbors in a specific peer group. The <as-num> parameter specifies all neighbors within the specified AS. The **all** parameter specifies all neighbors.

Chapter 39

Securing Access to Management Functions

This chapter explains how to secure access to management functions on a Foundry device.

NOTE: For all Foundry devices, RADIUS Challenge is supported for 802.1x authentication but not for login authentication. Also, multiple challenges are supported for TACACS+ login authentication.

Securing Access Methods

The following table lists the management access methods available on a Foundry device, how they are secured by default, and the ways in which they can be secured.

Table 39.1: Ways to secure management access to Foundry devices

Access method	How the access method is secured by default	Ways to secure the access method	See page
Serial access to the CLI	Not secured	Establish passwords for management privilege levels	39-13
Access to the Privileged EXEC and CONFIG levels of the CLI	Not secured	Establish a password for Telnet access to the CLI	39-13
		Establish passwords for management privilege levels	39-13
		Set up local user accounts	39-17
		Configure TACACS/TACACS+ security	39-27
		Configure RADIUS security	39-42

Table 39.1: Ways to secure management access to Foundry devices (Continued)

Access method	How the access method is secured by default	Ways to secure the access method	See page
Telnet access	Not secured	Regulate Telnet access using ACLs	39-4
		Allow Telnet access only from specific IP addresses	39-6
		Restrict Telnet access based on a client’s MAC address	39-7
		Allow Telnet access only from specific MAC addresses	39-9
		Specify the maximum number of login attempts for Telnet access	39-8
		Disable Telnet access	39-12
		Establish a password for Telnet access	39-13
		Establish passwords for privilege levels of the CLI	39-13
		Set up local user accounts	39-17
		Configure TACACS/TACACS+ security	39-27
		Configure RADIUS security	39-42
Secure Shell (SSH) access	Not configured	Configure SSH	40-1
		Regulate SSH access using ACLs	39-5
		Allow SSH access only from specific IP addresses	39-7
		Allow SSH access only from specific MAC addresses	39-7
		Establish passwords for privilege levels of the CLI	39-13
		Set up local user accounts	39-17
		Configure TACACS/TACACS+ security	39-27
		Configure RADIUS security	39-42

Table 39.1: Ways to secure management access to Foundry devices (Continued)

Access method	How the access method is secured by default	Ways to secure the access method	See page
Web management access	SNMP read or read-write community strings	Regulate Web management access using ACLs	39-5
		Allow Web management access only from specific IP addresses	39-7
		Allow Web management access only to clients connected to a specific VLAN	39-9
		Disable Web management access	39-12
		Configure SSL security for the Web management interface	39-24
		Set up local user accounts	39-17
		Establish SNMP read or read-write community strings for SNMP versions 1 and 2	47-1
		Establishing user groups for SNMP version 3	47-6
		Configure TACACS/TACACS+ security	39-27
		Configure RADIUS security	39-42
SNMP (IronView Network Manager) access	SNMP read or read-write community strings and the password to the Super User privilege level **Note**: SNMP read or read-write community strings are always required for SNMP access to the device.	Regulate SNMP access using ACLs	39-5
		Allow SNMP access only from specific IP addresses	39-7
		Disable SNMP access	39-12
		Allow SNMP access only to clients connected to a specific VLAN	39-9
		Establish passwords to management levels of the CLI	39-13
		Set up local user accounts	39-17
		Establish SNMP read or read-write community strings	39-27
TFTP access	Not secured	Allow TFTP access only to clients connected to a specific VLAN	39-9
		Disable TFTP access	39-13

Restricting Remote Access to Management Functions

You can restrict access to management functions from remote sources, including Telnet, the Web management interface, and SNMP. The following methods for restricting remote access are supported:

- Using ACLs to restrict Telnet, Web management interface, or SNMP access
- Allowing remote access only from specific IP addresses
- Allowing Telnet and SSH access only from specific MAC addresses

- Allowing remote access only to clients connected to a specific VLAN
- Specifically disabling Telnet, Web management interface, or SNMP access to the device

The following sections describe how to restrict remote access to a Foundry device using these methods.

Using ACLs to Restrict Remote Access

You can use standard ACLs to control the following access methods to management functions on a Foundry device:

- Telnet
- SSH
- Web management
- SNMP

To configure access control for these management access methods:

1. Configure an ACL with the IP addresses you want to allow to access the device.
2. Configure a Telnet access group, SSH access group, web access group, and SNMP community strings. Each of these configuration items accepts an ACL as a parameter. The ACL contains entries that identify the IP addresses that can use the access method.

The following sections present examples of how to secure management access using ACLs. See the "Configuring Rule-Based IP Access Control Lists (ACLs)" on page 17-1 for more information on configuring ACLs.

Using an ACL to Restrict Telnet Access

To configure an ACL that restricts Telnet access to the device, enter commands such as the following:

```
FastIron(config)#access-list 10 deny host 209.157.22.32 log
FastIron(config)#access-list 10 deny 209.157.23.0 0.0.0.255 log
FastIron(config)#access-list 10 deny 209.157.24.0 0.0.0.255 log
FastIron(config)#access-list 10 deny 209.157.25.0/24 log
FastIron(config)#access-list 10 permit any
FastIron(config)#telnet access-group 10
FastIron(config)#write memory
```

Syntax: telnet access-group <num>

The <num> parameter specifies the number of a standard ACL and must be from 1 – 99.

The commands above configure ACL 10, then apply the ACL as the access list for Telnet access. The device allows Telnet access to all IP addresses except those listed in ACL 10.

To configure a more restrictive ACL, create permit entries and omit the **permit any** entry at the end of the ACL. For example:

```
FastIron(config)#access-list 10 permit host 209.157.22.32
FastIron(config)#access-list 10 permit 209.157.23.0 0.0.0.255
FastIron(config)#access-list 10 permit 209.157.24.0 0.0.0.255
FastIron(config)#access-list 10 permit 209.157.25.0/24
FastIron(config)#telnet access-group 10
FastIron(config)#write memory
```

The ACL in this example permits Telnet access only to the IP addresses in the **permit** entries and denies Telnet access from all other IP addresses.

Using an ACL to Restrict SSH Access

To configure an ACL that restricts SSH access to the device, enter commands such as the following:

```
FastIron(config)#access-list 12 deny host 209.157.22.98 log
FastIron(config)#access-list 12 deny 209.157.23.0 0.0.0.255 log
FastIron(config)#access-list 12 deny 209.157.24.0/24 log
FastIron(config)#access-list 12 permit any
FastIron(config)#ssh access-group 12
FastIron(config)#write memory
```

Syntax: ssh access-group <num>

The <num> parameter specifies the number of a standard ACL and must be from 1 – 99.

These commands configure ACL 12, then apply the ACL as the access list for SSH access. The device denies SSH access from the IP addresses listed in ACL 12 and permits SSH access from all other IP addresses. Without the last ACL entry for permitting all packets, this ACL would deny SSH access from all IP addresses.

NOTE: In this example, the command **ssh access-group 10** could have been used to apply the ACL configured in the example for Telnet access. You can use the same ACL multiple times.

Using an ACL to Restrict Web Management Access

To configure an ACL that restricts Web management access to the device, enter commands such as the following:

```
FastIron(config)#access-list 12 deny host 209.157.22.98 log
FastIron(config)#access-list 12 deny 209.157.23.0 0.0.0.255 log
FastIron(config)#access-list 12 deny 209.157.24.0/24 log
FastIron(config)#access-list 12 permit any
FastIron(config)#web access-group 12
FastIron(config)#write memory
```

Syntax: web access-group <num>

The <num> parameter specifies the number of a standard ACL and must be from 1 – 99.

These commands configure ACL 12, then apply the ACL as the access list for Web management access. The device denies Web management access from the IP addresses listed in ACL 12 and permits Web management access from all other IP addresses. Without the last ACL entry for permitting all packets, this ACL would deny Web management access from all IP addresses.

Using ACLs to Restrict SNMP Access

To restrict SNMP access to the device using ACLs, enter commands such as the following:

NOTE: The syntax for using ACLs for SNMP access is different from the syntax for controlling Telnet, SSH, and Web management access using ACLs.

```
FastIron(config)#access-list 25 deny host 209.157.22.98 log
FastIron(config)#access-list 25 deny 209.157.23.0 0.0.0.255 log
FastIron(config)#access-list 25 deny 209.157.24.0 0.0.0.255 log
FastIron(config)#access-list 25 permit any
FastIron(config)#access-list 30 deny 209.157.25.0 0.0.0.255 log
FastIron(config)#access-list 30 deny 209.157.26.0/24 log
FastIron(config)#access-list 30 permit any
FastIron(config)#snmp-server community public ro 25
FastIron(config)#snmp-server community private rw 30
FastIron(config)#write memory
```

Syntax: snmp-server community <string> ro | rw <num>

The <string> parameter specifies the SNMP community string the user must enter to gain SNMP access.

The **ro** parameter indicates that the community string is for read-only ("get") access. The **rw** parameter indicates the community string is for read-write ("set") access.

The <num> parameter specifies the number of a standard ACL and must be from 1 – 99.

These commands configure ACLs 25 and 30, then apply the ACLs to community strings.

ACL 25 is used to control read-only access using the "public" community string. ACL 30 is used to control read-write access using the "private" community string.

When **snmp-server community** is configured, all incoming SNMP packets are validated first by their community strings and then by their bound ACLs.

Restricting Remote Access to the Device to Specific IP Addresses

By default, a Foundry device does not control remote management access based on the IP address of the managing device. You can restrict remote management access to a single IP address for the following access methods:

- Telnet access
- SSH access
- Web management access
- SNMP access

In addition, you can restrict all access methods to the same IP address using a single command.

The following examples show the CLI commands for restricting remote access. You can specify only one IP address with each command. However, you can enter each command ten times to specify up to ten IP addresses.

NOTE: You cannot restrict remote management access using the Web management interface.

Restricting Telnet Access to a Specific IP Address

To allow Telnet access to the Foundry device only to the host with IP address 209.157.22.39, enter the following command:

```
FastIron(config)#telnet-client 209.157.22.39
```

Syntax: [no] telnet-client <ip-addr> | <ipv6-addr>

Restricting SSH Access to a Specific IP Address

To allow SSH access to the Foundry device only to the host with IP address 209.157.22.39, enter the following command:

```
FastIron(config)#ip ssh client 209.157.22.39
```

Syntax: [no] ip ssh client <ip-addr> | <ipv6-addr>

Restricting Web Management Access to a Specific IP Address

To allow Web management access to the Foundry device only to the host with IP address 209.157.22.26, enter the following command:

```
FastIron(config)#web-client 209.157.22.26
```

Syntax: [no] web-client <ip-addr> | <ipv6-addr>

Restricting SNMP Access to a Specific IP Address

To allow SNMP access (which includes IronView Network Manager) to the Foundry device only to the host with IP address 209.157.22.14, enter the following command:

```
FastIron(config)#snmp-client 209.157.22.14
```

Syntax: [no] snmp-client <ip-addr> | <ipv6-addr>

Restricting All Remote Management Access to a Specific IP Address

To allow Telnet, Web, and SNMP management access to the Foundry device only to the host with IP address 209.157.22.69, enter three separate commands (one for each access type) or enter the following command:

```
FastIron(config)#all-client 209.157.22.69
```

Syntax: [no] all-client <ip-addr> | <ipv6-addr>

Restricting Access to the Device Based on IP or MAC Address

Platform Support:

- FESX/FSX/FWSX devices running software release 03.0.00 and later
- FGS devices running software release 02.5.00 and later
- FLS devices running software release 03.0.00 and later

You can restrict remote management access to the Foundry device, using Telnet, SSH, HTTP, and HTTPS, based on the connecting client's IP or MAC address.

Restricting Telnet Connection

You can restrict Telnet connection to a device based on the client's IP address or MAC address.

To allow Telnet access to the Foundry device only to the host with IP address 209.157.22.39 ***and*** MAC address 0007.e90f.e9a0, enter the following command:

```
FastIron(config)#telnet client 209.157.22.39 0007.e90f.e9a0
```

Syntax: [no] telnet client <ip-addr> | <ipv6-addr> <mac-addr>

The following command allows Telnet access to the Foundry device to a host with any IP address and MAC address 0007.e90f.e9a0:

```
FastIron(config)#telnet client any 0007.e90f.e9a0
```

Syntax: [no] telnet client any <mac-addr>

Restricting SSH Connection

Platform Support:

- FastIron X-Series devices running software version 03.0.00 and later

- FGS and FLS devices running software release 02.5.00 and later
- FLS devices running software release 03.0.00 and later

You can restrict SSH connection to a device based on the client’s IP address or MAC address.

To allow SSH access to the Foundry device only to the host with IP address 209.157.22.39 ***and*** MAC address 0007.e90f.e9a0, enter the following command:

```
FastIron(config)#ip ssh client 209.157.22.39 0007.e90f.e9a0
```

Syntax: [no] ip ssh client <ip-addr> | <ipv6-addr> <mac-addr>

To allow SSH access to the Foundry device to a host with any IP address and MAC address 0007.e90f.e9a0, enter the following command:

```
FastIron(config)#ip ssh client any 0007.e90f.e9a0
```

Syntax: [no] ip ssh client any <mac-addr>

Restricting HTTP and HTTPS Connection

Platform Support:

- FGS and FLS devices running software release 02.6.00 and later
- FESX/FSX/FWSX devices running software release 03.3.00 and later

You can restrict an HTTP or HTTPS connection to a device based on the client’s IP address or MAC address.

To allow HTTP and HTTPS access to the Foundry device only to the host with IP address 209.157.22.40 ***and*** MAC address 0007.e90f.ab1c, enter the following command:

```
FastIron(config)#web client 209.157.22.40 0007.e90f.ab1c
```

Syntax: [no] web client <ip-addr> | <ipv6-addr> <mac-addr>

The following command allows HTTP and HTTPS access to the Foundry device to a host with any IP address and MAC address 0007.e90f.10ba:

```
FastIron(config)#web client any 0007.e90f.10ba
```

Syntax: [no] web client any <mac-addr>

Specifying the Maximum Number of Login Attempts for Telnet Access

If you are connecting to the Foundry device using Telnet, the device prompts you for a username and password. By default, you have up to 4 chances to enter a correct username and password. If you do not enter a correct username or password after 4 attempts, the Foundry device disconnects the Telnet session.

You can specify the number of attempts a Telnet user has to enter a correct username and password before the device disconnects the Telnet session. For example, to allow a Telnet user up to 5 chances to enter a correct username and password, enter the following command:

```
FastIron(config)#telnet login-retries 5
```

Syntax: [no] telnet login-retries <number>

You can specify from 0 – 5 attempts. The default is 4 attempts.

Restricting Remote Access to the Device to Specific VLAN IDs

You can restrict management access to a Foundry device to ports within a specific port-based VLAN. VLAN-based access control applies to the following access methods:

- Telnet access
- Web management access
- SNMP access

- TFTP access

By default, access is allowed for all the methods listed above on all ports. Once you configure security for a given access method based on VLAN ID, access to the device using that method is restricted to only the ports within the specified VLAN.

VLAN-based access control works in conjunction with other access control methods. For example, suppose you configure an ACL to permit Telnet access only to specific client IP addresses, and you also configure VLAN-based access control for Telnet access. In this case, the only Telnet clients that can access the device are clients that have one of the IP addresses permitted by the ACL ***and*** are connected to a port that is in a permitted VLAN. Clients who have a permitted IP address but are connected to a port in a VLAN that is not permitted still cannot access the device through Telnet.

Restricting Telnet Access to a Specific VLAN

To allow Telnet access only to clients in a specific VLAN, enter a command such as the following:

```
FastIron(config)#telnet server enable vlan 10
```

The command in this example configures the device to allow Telnet management access only to clients connected to ports within port-based VLAN 10. Clients connected to ports that are not in VLAN 10 are denied management access.

Syntax: [no] telnet server enable vlan <vlan-id>

Restricting Web Management Access to a Specific VLAN

To allow Web management access only to clients in a specific VLAN, enter a command such as the following:

```
FastIron(config)#web-management enable vlan 10
```

The command in this example configures the device to allow Web management access only to clients connected to ports within port-based VLAN 10. Clients connected to ports that are not in VLAN 10 are denied management access.

Syntax: [no] web-management enable vlan <vlan-id>

Restricting SNMP Access to a Specific VLAN

To allow SNMP access only to clients in a specific VLAN, enter a command such as the following:

```
FastIron(config)#snmp-server enable vlan 40
```

The command in this example configures the device to allow SNMP access only to clients connected to ports within port-based VLAN 40. Clients connected to ports that are not in VLAN 40 are denied access.

Syntax: [no] snmp-server enable vlan <vlan-id>

Restricting TFTP Access to a Specific VLAN

To allow TFTP access only to clients in a specific VLAN, enter a command such as the following:

```
FastIron(config)#tftp client enable vlan 40
```

The command in this example configures the device to allow TFTP access only to clients connected to ports within port-based VLAN 40. Clients connected to ports that are not in VLAN 40 are denied access.

Syntax: [no] tftp client enable vlan <vlan-id>

Designated VLAN for Telnet Management Sessions to a Layer 2 Switch

By default, the management IP address you configure on a Layer 2 Switch applies globally to all the ports on the device. This is true even if you divide the device's ports into multiple port-based VLANs.

If you want to restrict the IP management address to a specific port-based VLAN, you can make that VLAN the designated management VLAN for the device. When you configure a VLAN to be the designated management VLAN, the management IP address you configure on the device is associated only with the ports in the designated VLAN. To establish a Telnet management session with the device, a user must access the device through one of the ports in the designated VLAN.

You also can configure up to five default gateways for the designated VLAN, and associate a metric with each one. The software uses the gateway with the lowest metric. The other gateways reside in the configuration but are not used. To use one of the other gateways, modify the configuration so that the gateway you want to use has the lowest metric.

If more than one gateway has the lowest metric, the gateway that appears first in the running-config is used.

NOTE: If you have already configured a default gateway globally and you do not configure a gateway in the VLAN, the software uses the globally configured gateway and gives the gateway a metric value of 1.

To configure a designated management VLAN, enter commands such as the following:

```
FastIron(config)#vlan 10 by port
FastIron(config-vlan-10)#untag ethernet 1/1 to 1/4
FastIron(config-vlan-10)#management-vlan
FastIron(config-vlan-10)#default-gateway 10.10.10.1 1
FastIron(config-vlan-10)#default-gateway 20.20.20.1 2
```

These commands configure port-based VLAN 10 to consist of ports 1/1 – 1/4 and to be the designated management VLAN. The last two commands configure default gateways for the VLAN. Since the 10.10.10.1 gateway has a lower metric, the software uses this gateway. The other gateway remains in the configuration but is not used. You can use the other one by changing the metrics so that the 20.20.20.1 gateway has the lower metric.

Syntax: [no] management-vlan

Syntax: [no] default-gateway <ip-addr> <metric>

The <ip-addr> parameters specify the IP address of the gateway router.

The <metric> parameter specifies the metric (cost) of the gateway. You can specify a value from 1 – 5. There is no default. The software uses the gateway with the lowest metric.

Device Management Security

By default, all management access is disabled. Each of the following management access methods must be specifically enabled as required in your installation:

- Telnet
- SSHv2
- SNMP
- Web management through HTTP
- Web management through HTTPS

The commands for granting access to each of these management interfaces is described in the following:

Telnet

To allow Telnet access to the Foundry device, enter the following command:

```
FastIron(config)#telnet server
```

Syntax: [no] telnet server

SSHv2

To allow SSHv2 access to the Foundry device, you must generate a Crypto Key as shown in the following command:

```
FastIron(config)#crypto key generate
```

Syntax: crypto key [generate | zeroize]

The **generate** parameter generates a dsa key pair.

The **zeroize** parameter deletes the currently operative dsa key pair.

In addition, you must use AAA authentication to create a password to allow SSHv2 access. For example the following command configures AAA authentication to use TACACS+ for authentication as the default or local if TACACS+ isn't available.

```
FastIron(config)#aaa authentication login default tacacs+
```

SNMP

To allow SNMP access to the Foundry device, enter the following command:

```
FastIron(config)#snmp-server
```

Syntax: [no] snmp-server

Web Management through HTTP

To allow web management through HTTP for the Foundry device, you enable web management as shown in the following command:

```
FastIron(config)#web-management
```

Syntax: [no] web-management [http | https]

Using the web-management command without the http or https option makes web management available for both.

The **http** option specifies that web management is enabled for HTTP access.

The **https** option specifies that web management is enabled for HTTPS access.

Web Management through HTTPS

To allow web management through HTTPS, you must enable web management as shown in "Web Management through HTTP" . Additionally, you must generate a crypto SSL certificate or import digital certificates issued by a third-party Certificate Authority (CA).

To generate a crypto SSL certificate use the following command:

```
FastIron(config)#crypto-ssl certificate generate
```

Syntax: crypto-ssl certificate [generate | zeroize]

Using the web-management command without the http or https option makes web management available for both.

The **generate** parameter generates an ssl certificate.

The **zeroize** parameter deletes the currently operative ssl certificate.

To import a digital certificate issued by a third-party Certificate Authority (CA) and save it in the flash memory, use the following command:

```
FastIron(config)#ip ssl certificate-data-file tftp 10.10.10.1 cacert.pem
```

Syntax: ip ssl certificate-data-file tftp <ip-addr> <file-name>

The <ip-addr> variable is the IP address of the TFTP server from which the digital certificate file is being downloaded.

The <file-name> variable is the file name of the digital certificate that you are importing to the router.

Disabling Specific Access Methods

You can specifically disable the following access methods:

- Telnet access
- Web management access
- SNMP access
- TFTP

NOTE: If you disable Telnet access, you will not be able to access the CLI except through a serial connection to the management module. If you disable SNMP access, you will not be able to use IronView Network Manager or third-party SNMP management applications.

Disabling Telnet Access

You can use a Telnet client to access the CLI on the device over the network. If you do not plan to use the CLI over the network and want to disable Telnet access to prevent others from establishing CLI sessions with the device, enter the following command:

```
FastIron(config)#no telnet server
```

To re-enable Telnet operation, enter the following command:

```
FastIron(config)#telnet server
```

Syntax: [no] telnet server

Disabling Web Management Access

If you want to prevent access to the device through the Web management interface, you can disable the Web management interface.

NOTE: As soon as you make this change, the device stops responding to Web management sessions. If you make this change using your Web browser, your browser can contact the device, but the device will not reply once the change takes place.

To disable the Web management interface, enter the following command:

```
FastIron(config)#no web-management
```

To re-enable the Web management interface, enter the following command:

```
FastIron(config)#web-management
```

Syntax: [no] web-management

Disabling Web Management Access by HP ProCurve Manager

By default, TCP ports 80 and 280 are enabled on the Foundry device. TCP port 80 (HTTP) allows access to the device’s Web management interface. TCP port 280 allows access to the device by HP ProCurve Manager.

The **no web-management** command disables both TCP ports. However, if you want to disable only port 280 and leave port 80 enabled, use the **hp-top-tools** option with the command. Here is an example.

```
FastIron(config)#no web-management hp-top-tools
```

Syntax: [no] web-management [allow-no-password | enable [vlan <vlan-id>] | front-panel | hp-top-tools | list-menu]

The **hp-top-tools** parameter disables TCP port 280.

Disabling SNMP Access

SNMP is required if you want to manage a Foundry *device using* IronView Network Manager*.*

To disable SNMP management of the device:

```
FastIron(config)#no snmp-server
```

To later re-enable SNMP management of the device:

```
FastIron(config)#snmp-server
```

Syntax: no snmp-server

Disabling TFTP Access

You can globally disable TFTP to block TFTP client access. By default, TFTP client access is enabled.

To disable TFTP client access, enter the following command at the Global CONFIG level of the CLI:

```
FastIron(config)#tftp disable
```

When TFTP is disabled, users are prohibited from using the **copy tftp** command to copy files to the system flash. If users enter this command while TFTP is disabled, the system will reject the command and display an error message.

To re-enable TFTP client access once it is disabled, enter the following command:

```
FastIron(config)#no tftp disable
```

Syntax: [no] tftp disable

Setting Passwords

Passwords can be used to secure the following access methods:

- Telnet access can be secured by setting a Telnet password. See “Setting a Telnet Password” on page 39-13.
- Access to the Privileged EXEC and CONFIG levels of the CLI can be secured by setting passwords for management privilege levels. See “Setting Passwords for Management Privilege Levels” on page 39-13.

This section also provides procedures for enhancing management privilege levels, recovering from a lost password, and disabling password encryption.

NOTE: You also can configure up to 16 user accounts consisting of a user name and password, and assign each user account a management privilege level. See “Setting up Local User Accounts” on page 39-17.

Setting a Telnet Password

By default, the device does not require a user name or password when you log in to the CLI using Telnet. You can assign a password for Telnet access using one of the following methods.

Set the password “letmein” for Telnet access to the CLI using the following command at the global CONFIG level:

```
FastIron(config)#enable telnet password letmein
```

Syntax: [no] enable telnet password <string>

Suppressing Telnet Connection Rejection Messages

By default, if a Foundry device denies Telnet management access to the device, the software sends a message to the denied Telnet client. You can optionally suppress the rejection message. When you enable the option, a denied Telnet client does not receive a message from the Foundry device. Instead, the denied client simply does not gain access.

To suppress the connection rejection message, use the following CLI method.

To suppress the connection rejection message sent by the device to a denied Telnet client, enter the following command at the global CONFIG level of the CLI:

```
FastIron(config)#telnet server suppress-reject-message
```

Syntax: [no] telnet server suppress-reject-message

Setting Passwords for Management Privilege Levels

You can set one password for each of the following management privilege levels:

- Super User level – Allows complete read-and-write access to the system. This is generally for system administrators and is the only management privilege level that allows you to configure passwords.

- Port Configuration level – Allows read-and-write access for specific ports but not for global (system-wide) parameters.
- Read Only level – Allows access to the Privileged EXEC mode and CONFIG mode of the CLI but only with read access.

You can assign a password to each management privilege level. You also can configure up to 16 user accounts consisting of a user name and password, and assign each user account to one of the three privilege levels. See "Setting up Local User Accounts" on page 39-17.

NOTE: You must use the CLI to assign a password for management privilege levels. You cannot assign a password using the Web management interface.

If you configure user accounts in addition to privilege level passwords, the device will validate a user's access attempt using one or both methods (local user account or privilege level password), depending on the order you specify in the authentication-method lists. See "Configuring Authentication-Method Lists" on page 39-57.

To set passwords for management privilege levels:

1. At the opening CLI prompt, enter the following command to change to the Privileged level of the EXEC mode:

   ```
   FastIron> enable
   FastIron#
   ```

2. Access the CONFIG level of the CLI by entering the following command:

   ```
   FastIron#configure terminal
   FastIron(config)#
   ```

3. Enter the following command to set the Super User level password:

   ```
   FastIron(config)#enable super-user-password <text>
   ```

   NOTE: You must set the Super User level password before you can set other types of passwords. The Super User level password can be an alphanumeric string, but cannot begin with a number.

4. Enter the following commands to set the Port Configuration level and Read Only level passwords:

   ```
   FastIron(config)#enable port-config-password <text>
   FastIron(config)#enable read-only-password <text>
   ```

Syntax: `enable super-user-password` <text>

Syntax: `enable port-config-password` <text>

Syntax: `enable read-only-password` <text>

NOTE: If you forget your Super User level password, see "Recovering from a Lost Password" on page 39-16.

Augmenting Management Privilege Levels

Each management privilege level provides access to specific areas of the CLI by default:

- Super User level provides access to all commands and displays.
- Port Configuration level gives access to:
  - The User EXEC and Privileged EXEC levels
  - The port-specific parts of the CONFIG level
  - All interface configuration levels
- Read Only level gives access to:

- The User EXEC and Privileged EXEC levels

You can grant additional access to a privilege level on an individual command basis. To grant the additional access, you specify the privilege level you are enhancing, the CLI level that contains the command, and the individual command.

NOTE: This feature applies only to management privilege levels on the CLI. You cannot augment management access levels for the Web management interface.

Enhance the Port Configuration privilege level so users also can enter IP commands at the global CONFIG level:

```
FastIron(config)#privilege configure level 4 ip
```

In this command, **configure** specifies that the enhanced access is for a command at the global CONFIG level of the CLI. The **level 4** parameter indicates that the enhanced access is for management privilege level 4 (Port Configuration). All users with Port Configuration privileges will have the enhanced access. The **ip** parameter indicates that the enhanced access is for the IP commands. Users who log in with valid Port Configuration level user names and passwords can enter commands that begin with "ip" at the global CONFIG level.

Syntax: [no] privilege <cli-level> level <privilege-level> <command-string>

The <cli-level> parameter specifies the CLI level and can be one of the following values:

- **exec** – EXEC level; for example, FastIron> or FastIron#
- **configure** – CONFIG level; for example, FastIron`(config)#`
- **interface** – Interface level; for example, FastIron`(config-if-6)#`
- **loopback-interface** – loopback interface level
- **virtual-interface** – Virtual-interface level; for example, FastIron`(config-vif-6)#`
- **dot1x** – 802.1X configuration level
- **ipv6-access-list** – IPv6 access list configuration level
- **rip-router** – RIP router level; for example, FastIron`(config-rip-router)#`
- **ospf-router** – OSPF router level; for example, FastIron`(config-ospf-router)#`
- **dvmrp-router** – DVMRP router level; for example, FastIron`(config-dvmrp-router)#`
- **pim-router** – PIM router level; for example, FastIron`(config-pim-router)#`
- **bgp-router** – BGP4 router level; for example, FastIron`(config-bgp-router)#`
- **vrrp-router** – VRRP configuration level
- **gvrp** – GVRP configuration level
- **trunk** – trunk configuration level
- **port-vlan** – Port-based VLAN level; for example, FastIron`(config-vlan)#`
- **protocol-vlan** – Protocol-based VLAN level

The <privilege-level> indicates the number of the management privilege level you are augmenting. You can specify one of the following:

- **0** – Super User level (full read-write access)
- **4** – Port Configuration level
- **5** – Read Only level

The <command-string> parameter specifies the command you are allowing users with the specified privilege level to enter. To display a list of the commands at a CLI level, enter "?" at that level's command prompt.

Recovering from a Lost Password

Recovery from a lost password requires direct access to the serial port and a system reset.

NOTE: You can perform this procedure only from the CLI.

To recover from a lost password:

1. Start a CLI session over the serial interface to the device.
2. Reboot the device.
3. At the initial boot prompt at system startup, enter **b** to enter the boot monitor mode.
4. Enter **no password** at the prompt. (You cannot abbreviate this command.) This command will cause the device to bypass the system password check.
5. Enter **boot system flash primary** at the prompt.
6. After the console prompt reappears, assign a new password.

Displaying the SNMP Community String

If you want to display the SNMP community string, enter the following commands:

```
FastIron(config)#enable password-display
FastIron#show snmp server
```

The **enable password-display** command enables display of the community string, but only in the output of the **show snmp server** command. Display of the string is still encrypted in the startup-config file and running-config. Enter the command at the global CONFIG level of the CLI.

Disabling Password Encryption

When you configure a password, then save the configuration to the Foundry device's flash memory, the password is also saved to flash as part of the configuration file. By default, the passwords are encrypted so that the passwords cannot be observed by another user who displays the configuration file. Even if someone observes the file while it is being transmitted over TFTP, the password is encrypted.

NOTE: You cannot disable password encryption using the Web management interface.

If you want to remove the password encryption, you can disable encryption by entering the following command:

```
FastIron(config)#no service password-encryption
```

Syntax: [no] service password-encryption

Specifying a Minimum Password Length

By default, the Foundry device imposes no minimum length on the Line (Telnet), Enable, or Local passwords. You can configure the device to require that Line, Enable, and Local passwords be at least a specified length.

For example, to specify that the Line, Enable, and Local passwords be at least 8 characters, enter the following command:

```
FastIron(config)#enable password-min-length 8
```

Syntax: enable password-min-length <number-of-characters>

The <number-of-characters> can be from 1 – 48.

Setting up Local User Accounts

You can define up to 16 local user accounts on a Foundry device. User accounts regulate who can access the management functions in the CLI using the following methods:

- Telnet access
- Web management access
- SNMP access

NOTE: Local user accounts are not supported on the FastIron Workgroup Layer 2 Switch.

Local user accounts provide greater flexibility for controlling management access to Foundry devices than do management privilege level passwords and SNMP community strings of SNMP versions 1 and 2. You can continue to use the privilege level passwords and the SNMP community strings as additional means of access authentication. Alternatively, you can choose not to use local user accounts and instead continue to use only the privilege level passwords and SNMP community strings. Local user accounts are backward-compatible with configuration files that contain privilege level passwords. See “Setting Passwords for Management Privilege Levels” on page 39-13.

If you configure local user accounts, you also need to configure an authentication-method list for Telnet access, Web management access, and SNMP access. See “Configuring Authentication-Method Lists” on page 39-57.

For each local user account, you specify a user name. You also can specify the following parameters:

- A password
- A management privilege level, which can be one of the following:
  - Super User level (default) – Allows complete read-and-write access to the system. This is generally for system administrators and is the only privilege level that allows you to configure passwords.
  - Port Configuration level – Allows read-and-write access for specific ports but not for global parameters.
  - Read Only level – Allows access to the Privileged EXEC mode and CONFIG mode with read access only.
- Starting in release 03.0.00 for the FastIron X Series devices, you can set additional username and password rules. See “Enhancements to Username and Password” .

Enhancements to Username and Password

Platform Support:

- FastIron X Series devices running software release 03.0.00 or later
- FGS and FLS devices running software release 03.2.00 and later

This section describes the enhancements to the username and password features introduced in the releases listed above.

The following rules are enabled by default:

- Users are required to accept the message of the day.
- Users are locked out (disabled) if they fail to login after three attempts. This feature is automatically enabled in release 03.0.00 for FastIron X Series devices, and release 03.2.00 for FastIron GS devices. Use the **disable-on-login-failure** command to change the number of login attempts (up to 10) before users are locked out.

The following rules are disabled by default:

- Enhanced user password combination requirements
- User password masking
- Quarterly updates of user passwords

- You can configure the system to store up to 15 previously configured passwords for each user.
- You can use the **disable-on-login-failure** command to change the number of login attempts (up to 10) before users are locked out.
- A password can now be set to expire.

Enabling Enhanced User Password Combination Requirements

Platform Support:

- FESX/FSX/FWSX devices running software release 03.0.00 and later
- FGS and FLS devices running software release 03.2.00 and later

When strict password enforcement is enabled on the Foundry device, you must enter a minimum of eight characters containing the following combinations when you create an enable and a user password:

- At least two upper case characters
- At least two lower case characters
- At least two numeric characters
- At least two special characters

NOTE: Password minimum and combination requirements are strictly enforced.

Use the **enable strict-password-enforcement** command to enable the password security feature.

```
FastIron(config)#enable strict-password-enforcement
```

Syntax: [no] enable strict-password-enforcement

This feature is disabled by default.

NOTE: When you upgrade to release FSX 03.0.00 or later or to FGS 03.2.00 or later, the old passwords are still valid; however, users must change their passwords to follow the new format to take advantage of this password enhancement.

The following security upgrades apply to the **enable strict-password-enforcement** command:

- Passwords must not share four or more concurrent characters with any other password configured on the router. If the user tries to create a password with four or more concurrent characters, the following error message will be returned:

  ```
  Error - The substring <str> within the password has been used earlier, please
  choose a different password.
  ```

  For example, the previous password was Ma!i4aYa&, the user cannot use any of the following as his or her new password:

  - Ma!imai$D because “Mail” were used consecutively in the previous password
  - &3B9aYa& because “aYa&” were used consecutively in the previous password
  - i4aYEv#8 because “i4aY“ were used consecutively in the previous password

- If the user tries to configure a password that was previously used, the Local User Account configuration will not be allowed and the following message will be displayed:

  ```
  This password was used earlier for same or different user, please choose a
  different password.
  ```

Enabling User Password Masking

By default, when you use the CLI to create a user password, the password displays on the console as you type it. For enhanced security, you can configure the Foundry device to mask the password characters entered at the CLI.

When password masking is enabled, the CLI displays asterisks (*) on the console instead of the actual password characters entered.

The following shows the default CLI behavior when configuring a username and password:

```
FastIron(config)#username kelly password summertime
```

The following shows the CLI behavior when configuring a username and password when **password-masking** is enabled:

```
FastIron(config)#username kelly password
Enter Password: ********
```

NOTE: When password masking is enabled, press the [Enter] key before entering the password.

Syntax: username <name> password [Enter]

For [Enter], press the Enter key. Enter the password when prompted.

If **strict-password-enforcement** is enabled, enter a password which contains the required character combination. See "Enabling Enhanced User Password Combination Requirements" on page 39-18.

To enable password masking, enter the following command:

```
FastIron(config)#enable user password-masking
```

Syntax: [no] enable user password-masking

Enabling User Password Aging

For enhanced security, password aging enforces quarterly updates of all user passwords. After 180 days, the CLI will automatically prompt users to change their passwords when they attempt to sign on.

When password aging is enabled, the software records the system time that each user password was configured or last changed. The time displays in the output of the **show running configuration** command, indicated by **set-time <time>**. For example:

```
FastIron#show run
Current configuration:
....
username waldo password .....
username raveen set-time 2086038248

....
```

The password aging feature uses the SNTP server clock to record the set-time. If the network does not have an SNTP server, then set-time will appear as **set-time 0** in the output of the **show running configuration** command.

A username's set-time configuration is removed when:

- The username and password is deleted from the configuration
- The username's password expires

When a username's set-time configuration is removed, it no longer appears in the **show running configuration** output.

Note that if a username does not have an assigned password, the username will not have a set-time configuration.

Password aging is disabled by default. To enable it, enter the following command at the global CONFIG level of the CLI:

```
FastIron(config)#enable user password-aging
```

Syntax: [no] enable user password-aging

Configuring Password History

Platform Support:

- FESX/FSX/FWSX devices running software release 03.0.00 and later
- FGS and FLS devices running software release 03.2.00 and later

By default, the Foundry device stores the last five user passwords for each user. When changing a user password, the user cannot use any of the five previously configured passwords.

For security purposes, you can configure the Foundry device to store up to 15 passwords for each user, so that users do not use the same password multiple times. If a user attempts to use a password that is stored, the system will prompt the user to choose a different password.

To configure enhanced password history, enter a command such as the following at the global CONFIG level of the CLI:

```
FastIron(config)#enable user password-history 15
```

Syntax: [no] enable user password-history <1 – 15>

Password History in Pre-Release 03.0.00 Software

The Foundry device stores not only the current password configured for a local user, but the previous two passwords configured for the user as well. The local user's password cannot be changed to one of the stored passwords.

Consequently, if you change the password for a local user, you must select a password that is different from the current password, as well as different from the previous two passwords that had been configured for that user.

For example, say local user waldo originally had a password of "whereis", and the password was subsequently changed to “whois”, then later changed to “whyis”. If you change waldo's password again, you cannot change it to "whereis", "whois", or "whyis".

The current and previous passwords are stored in the device’s running-config file in encrypted form. For example:

```
FastIron#show run
...
username waldo password 8 $1$Ro2..Ox0$udBu7pQT5XyuaXMUiUHy9. history
$1$eq...T62$IfpxIcxnDWX7CSVQKIodu. $1$QD3..2Q0$DYxgxCI64ZOSsYmSSaA28/
...
```

In the running-config file, the user’s previous two passwords are displayed in encrypted form following the **history** parameter.

Enhanced Login Lockout

Platform Support:

- FESX/FSX/FWSX devices running software release 03.0.00 and later
- FGS and FLS devices running software release 03.2.00 and later

The CLI provides up to three login attempts. If a user fails to login after three attempts, that user is locked out (disabled). If desired, you can increase or decrease the number of login attempts before the user is disabled. To do so, enter a command such as the following at the global CONFIG level of the CLI:

```
FastIron(config)#enable user disable-on-login-failure 7
```

Syntax: enable user disable-on-login-failure <1 – 10>

To re-enable a user that has been locked out, do one of the following:

- Reboot the Foundry device to re-enable all disabled users.
- Enable the user by entering the following command:

  ```
  FastIron(config)#username sandy enable
  ```

For example:

```
FastIron(config)#user sandy enable
FastIron#show user
Username  Password                           Encrypt   Priv Status   Expire Time
==============================================================================

sandy     $1$Gz...uX/$wQ44fVGtsqbKWkQknzAZ6. enabled   0   enabled  90 days
```

Syntax: username <name> enable

Setting Passwords to Expire

You can set a user password to expire. Once a password expires, the administrator must assign a new password to the user. To configure a user password to expire, enter the following:

```
FastIron(config)#username sandy expires 20
```

Syntax: username <name> expires <days>

Enter 1 – 365 for number of days. The default is 90 days.

For example:

```
FastIron(config)#username sandy expires 20
FastIron#show user
Username     Password                           Encrypt   Priv  Status   Expire Time
==============================================================================
sandy       $1$Gz...uX/$wQ44fVGtsqbKWkQknzAZ6. enabled    0    enabled  20 days
```

Requirement to Accept the Message of the Day

Platform Support:

- FESX/FSX/FWSX devices running software release 03.1.01a and later
- FGS/FLS devices running software release 03.2.00 and later

If a message of the day (MOTD) is configured, a user will be required to press the Enter key before he or she can login. MOTD is configured using the **banner motd** command.

There are no new CLI commands for this feature.

NOTE: Beginning with release 03.0.01a, this requirement is disabled by default, unless configured. Users are not required to press Enter after the MOTD banner is displayed. See “Requiring Users to Press the Enter Key after the Message of the Day Banner” on page 4-15.

Configuring a Local User Account

You can create accounts for local users with or without passwords. Accounts with passwords can have encrypted or unencrypted passwords.

You can assign privilege levels to local user accounts, but on a new device, you must create a local user account that has a Super User privilege before you can create accounts with other privilege levels.

NOTE: You must grant Super User level privilege to at least one account before you add accounts with other privilege levels. You need the Super User account to make further administrative changes.

Local User Accounts with No Passwords

To create a user account without a password, enter the following command at the global CONFIG level of the CLI:

```
FastIron(config)#username wonka nopassword
```

Syntax: [no] username <user-string> privilege <privilege-level> nopassword

Local User Accounts with Unencrypted Passwords

If you want to use unencrypted passwords for local user accounts, enter a command such as the following at the global CONFIG level of the CLI:

```
FastIron(config)#username wonka password willy
```

If password masking is enabled, press the [Enter] key before entering the password:

```
FastIron(config)#username wonka
Enter Password:  willy
```

The above commands add a local user account with the user name “wonka” and the password “willy”. This account has the Super User privilege level; this user has full access to all configuration and display features.

```
FastIron(config)#username waldo privilege 5 password whereis
```

This command adds a user account for user name “waldo”, password “whereis”, with the Read Only privilege level. Waldo can look for information but cannot make configuration changes.

Syntax: [no] username <user-string> privilege <privilege-level> password | nopassword <password-string>

You can enter up to 255 characters for <user-string>.

The **privilege** <privilege-level> parameter specifies the privilege level for the account. You can specify one of the following:

- **0** – Super User level (full read-write access)
- **4** – Port Configuration level
- **5** – Read Only level

The default privilege level is **0**. If you want to assign Super User level access to the account, you can enter the command without **privilege 0**, as shown in the command example above.

The **password | nopassword** parameter indicates whether the user must enter a password. If you specify **password**, enter the string for the user's password. You can enter up to 255 characters for <password-string>. If strict password enforcement is enabled on the device, you must enter a minimum of eight characters containing the following combinations:

- At least two upper case characters
- At least two lower case characters
- At least two numeric characters
- At least two special characters

NOTE: You must be logged on with Super User access (privilege level 0) to add user accounts or configure other access parameters.

To display user account information, enter the following command:

```
FastIron#show users
```

Syntax: show users

Local Accounts with Encrypted Passwords

You can create local user accounts with MD5 encrypted passwords using one of the following methods:

- Issuing the **service password-encryption** command after creating the local user account with a **username <user-string> [privilege <privilege-level>] password 0** command
- Using the **username** <user-string> **create-password** command (on FastIron X Series devices running software release 03.1.00 and later)

NOTE: To create an encrypted all-numeric password, use the **username** <user-string> **create-password** command.

If you create a local user account using the commands discussed in “Local User Accounts with Unencrypted Passwords” on page 39-22, you can issue the **service password-encryption** command to encrypt all passwords that have been previously entered.

For example,

```
FastIron(config)#username wonka privilege 5 password willy
FastIron(config)#service password-encryption
```

If password masking is enabled, enter the commands this way:

```
FastIron(config)#username wonka privilege 5 password
Enter Password: willy
FastIron(config)#service password-encryption
```

Syntax: [no] service password-encryption

Create Password Option

Platform Support:

- FESX/FSX/FWSX devices running software release 03.1.00 and later

As an alternative to the commands above, the **create-password** option allows you to create an encrypted password in one line of command. Also, this new option allows you to create an all-numeric, encrypted password.

You can enter:

```
FastIron(config)#username wonka privilege 5 create-password willy
```

Syntax: [no] username <user-string> [privilege <privilege-level>] create-password <password-string>

You can enter up to 255 characters for <user-string>. This string can be alphanumeric or all-numeric.

The **privilege** parameter specifies the privilege level for the account. You can specify one of the following:

- **0** – Super User level (full read-write access)
- **4** – Port Configuration level
- **5** – Read Only level

Enter up to 255 alphanumeric characters for <password-string>.

Changing a Local User Password

To change a local user password for an existing local user account, enter a command such as the following at the global CONFIG level of the CLI.

NOTE: You must be logged on with Super User access (privilege level 0) to change user passwords.

```
FastIron(config)#username wonka password willy
```

If password masking is enabled, enter the username, press the [Enter] key, then enter the password:

```
FastIron(config)#username wonka password
Enter Password:  willy
```

The above commands change wonka's user name password to “willy”.

Syntax: [no] username <user-string> password <password-string>

Enter up to 255 characters for <user-string>.

The <password-string> parameter is the user password. The password can be up to 255 characters and must differ from the current password and two previously configured passwords.

When a password is changed, a message such as the following is sent to the Syslog:

```
SYSLOG: <14>Jan 1 00:00:00 10.44.9.11 Security: Password has been changed for user
tester from console session.
```

The message includes the name of the user whose password was changed and during which session type, such as Console, Telnet, SSH, Web, SNMP, or others, the password was changed.

Configuring SSL Security for the Web Management Interface

The Foundry device supports Secure Sockets Layer / Transport Level Security (SSL 3.0 / TLS 1.0) for configuring the device using the Web Management interface.

When enabled, the SSL protocol uses digital certificates and public-private key pairs to establish a secure connection to the Foundry device. Digital certificates serve to prove the identity of a connecting client, and public-private key pairs provide a means to encrypt data sent between the device and the client.

Configuring SSL for the Web management interface consists of the following tasks:

- Optionally enabling the SSL server on the Foundry device

NOTE: The SSL server is automatically enabled when an SSL certificate is generated.

- Importing an RSA certificate and private key file from a client (optional)
- Generating a certificate

Enabling the SSL Server on the Foundry Device

To enable the SSL server on the Foundry device, enter the following command:

```
FastIron(config)#web-management https
```

Syntax: [no] web-management http | https

You can enable either the HTTP or HTTPs servers with this command. You can disable both the HTTP and HTTPs servers by entering the following command:

```
FastIron(config)#no web-management
```

Syntax: no web-management

Specifying a Port for SSL Communication

By default, SSL protocol exchanges occur on TCP port 443. You can optionally change the port number used for SSL communication.

For example, the following command causes the device to use TCP port 334 for SSL communication:

```
FastIron(config)#ip ssl port 334
```

Syntax: [no] ip ssl port <port-number>

The default port for SSL communication is 443.

Changing the SSL Server Certificate Key Size

The default key size for Foundry-issued and imported digital certificates is 1024 bits. If desired, you can change the default key size to a value between 512 and 4096 bits. To do so, enter a command such as the following at the Global CONFIG level of the CLI:

```
FastIron(config)#ip ssl cert-key-size 3000
```

Syntax: ip ssl cert-key-size <512 – 4096>

NOTE: The SSL server certificate key size applies to digital certificates issued by Foundry, as well as imported certificates.

Support for SSL Digital Certificates Larger than 2048 Bytes

Platform Support:

- FESX/FSX/FWSX devices running software release 04.0.00 and later

Devices running software release 04.0.00 and later have the ability to store and retrieve SSL digital certificates that are up to 4000 bytes in size. Previous releases support SSL certificates not larger than 2048 bytes.

To accomodate support for the larger SSL certificate file size, the SSL certificate is stored in flash memory, instead of in the configuration file. For backward compatibility, the Foundry device provides support as follows:

- If the SSL certificate size is less than 2048 bytes, the certificate will be written to the configuration file and to the flash file. This way, releases prior to FSX 04.0.00 can still store and retrieve the certificate.
- If the SSL certificate is larger than 2048 bytes, it will be written to the flash file only. Consequently, software versions prior to FSX 04.0.00 will not be able to store or retrieve an SSL certificate that is larger than 2048 bytes. In releases prior to FSX 04.0.00, if the certificate is larger than 2048 bytes, the following warning message will display on the console:

  ```
  Certificate is too big to store in the configuration file, this certificate works
  only with the released version later than 3.2.
  ```

Support for SSL certificates larger than 2048 bytes is automatically enabled in software release FSX 04.0.00 and later. You do not need to perform any configuration procedures to enable it.

Importing Digital Certificates and RSA Private Key Files

To allow a client to communicate with other Foundry device using an SSL connection, you configure a set of digital certificates and RSA public-private key pairs on the device. A digital certificate is used for identifying the connecting client to the server. It contains information about the issuing Certificate Authority, as well as a public key. You can either import digital certificates and private keys from a server, or you can allow the Foundry device to create them.

If you want to allow the Foundry device to create the digital certificates, see the next section, “Generating an SSL Certificate” . If you choose to import an RSA certificate and private key file from a client, you can use TFTP to transfer the files.

For example, to import a digital certificate using TFTP, enter a command such as the following:

```
FastIron(config)#ip ssl certificate-data-file tftp 192.168.9.210 certfile
```

Syntax: [no] ip ssl certificate-data-file tftp <ip-addr> <certificate-filename>

NOTE: If you are running a software version prior to 04.0.00 and you are importing a digital certificate from a client, it must be no larger than 2048 bytes. If you are running software release 04.0.00 or later, the digital certificate can be up to 4096 bytes. See “Support for SSL Digital Certificates Larger than 2048 Bytes” on page 39-25.

To import an RSA private key from a client using TFTP, enter a command such as the following:

```
FastIron(config)#ip ssl private-key-file tftp 192.168.9.210 keyfile
```

Syntax: [no] ip ssl private-key-file tftp <ip-addr> <key-filename>

The <ip-addr> is the IP address of a TFTP server that contains the digital certificate or private key.

Generating an SSL Certificate

After you have imported the digital certificate, it should automatically generate.

If the certificate does not automatically generate, enter the following command to generate it:

```
FastIron(config)#crypto-ssl certificate generate
```

Syntax: [no] crypto-ssl certificate generate

If you did not already import a digital certificate from a client, the device can create a default certificate. To do this, enter the following command:

```
FastIron(config)#crypto-ssl certificate generate default_cert
```

Syntax: [no] crypto-ssl certificate generate default_cert

Deleting the SSL Certificate

To delete the SSL certificate, enter the following command:

```
FastIron(config)#crypto-ssl certificate zeroize
```

Syntax: [no] crypto-ssl certificate zeroize

Viewing SSL Certificates

To view SSL certificates, use the **show ip ssl certificate** command. The following shows an example output.

```
FastIron#show ip ssl certificate

-----BEGIN CERTIFICATE-----
MIICqzCCAhSgAwIBAgIEfFWdFjANBgkqhkiG9w0BAQQFADBwMQswCQYD
VQQGEwJVUzETMBEGA1UECBMKQ2FsaWZvcm5pYTEYMBYGA1UEChMPVGhlIERlbW8gVmVuZG9yMRwwGgYDVQ
QLExNTZWN1cmUgRGV2
aWNlcyBEZXB0MRQwEgYDVQQDEwtEZW1vIFZlbmRvcjAeFw0wNDAxMDEwMDAwMDBaFw0xNjAxMDEwMDAwMD
BaMIGRMSAwHgYDVQQD
Exd3d3cuZm91bmRyeW5ldHdvcmtzLmNvbTEZMBcGA1UEChMQRm91bmRyeSBOZXR3b3JrczEkMCIGA1UECx
MbRm91bmRyeSBOZXR3
b3JrcyBDZXJpZmljYXRlMREwDwYDVQQHEwhTYW4gSm9zZTELMAkGA1UECBMCQ0ExDDAKBgNVBAYTA1VTQT
CBnzANBgkqhkiG9w0B
AQEFAAOBjQAwgYkCgYEAn0IwklktVj9L81fW5VmIVkMbdiyer1ukl/
SfMreAEw4BUlNTXllr7RM2Wdt2sQrk6MrTt9IK5iZ3eDg+
xwrIzB1++Zi5mGOC2/DcyeCmJXDWpZI/
r6VPspSvkDU4ulyumKC6FvQ8APWDD3SKwRsQuugQK7mzPNeTwqjUIOkuHL0CAwEAAaMw
MC4wHwYDVR0jBBgwFoAUQij9U3tWJXzFznovS3GqqmTw0B4wCwYDVR0PBAQDAgAgMA0GCSqGSIb3DQEBBA
UAA4GBAH3BMq1/KTJx
rGD9/zAVFr/XdddTKg2IX43G5bayNHbyPLCgVprcZ9SHlrLtYMB2H/5/rJEIg8jQtreFSYviaIs/
g4FRADj658bkjkNLrb3WwaQR
hRBDKUo7NoEkF9VfFNJiw1Bc4DciYe+oY7kRctm+KtUqMSvPuZD8SsUA2YL2MIICXQIBAAKBgQCfQjCSWS
1WP0vzV9blWYhWQxt2
LJ6vW6SX9J8yt4ATDgFSU1NeWWvtEzZZ23axCuToytO30grmJnd4OD7HCsjMHX75mLmYY4Lb8NzJ4KYlcN
alkj+vpU+ylK+QNTi6
XK6YoLoW9DwA9YMPdIrBGxC66BArubM815PCqNQg6S4cvQIDAQABAoGAJonK3S9NREiu5WUFMJzY3B8DHz
au87pWexsMBybWjEXg
Vf6p9vPmpQj4JwhvkWoUeKzUTPA8CPOYB3R85IecjT50TQ2313BpBhcXrRmqZMVBGjwXQVLKOwOTYpPcJz
98B0UNH8Uq3x5M+ITs
DgpmUaCHiZWVWmcqsgL6v9Z7TsECQQDQbRUb2uFccwRcrPmlWQiRD3ks6HcM57PeCXCNyPCAOj3OZA6Owi
fhMUx8eDirNtFvItNX
e/
ENs8gi+aoF5AIxAkEAw5wcRmyBhzEOBlk7N4xZ4aLtGVXHGTJCtTCOoMpYRiIlcBql+OzTGp0JLGV00RYb
JQA612TE3PxakxPm
yD+0TQJAOywc6BGWFZcZz+3T2luSkQxhjJxa4DEP4aOwbrBzhxQB5AyIDYOVqttCKbw/
6mvfvbuXYKjxYMB/u15CJPX8oQJBAI4r
GCHXGw03CgpzMCbfSzRDRi/
zuBEIBpPKBt+2MRJIHw2rUJXrCkN9fv1Vf+G3P2NVF28VPj+e6fWbsVA5fMECQQCpJBrD79i6obHt
07/gO+PimC6y1NFQtsoa/y8Mzwt8dzq0TrT0600r1fO2S2gbEzZxzYoGkfDX6WSuaBkj4zCt
-----END CERTIFICATE-----
```

Configuring TACACS/TACACS+ Security

You can use the security protocol Terminal Access Controller Access Control System (TACACS) or TACACS+ to authenticate the following kinds of access to the Foundry device

- Telnet access
- SSH access
- Web management access
- Access to the Privileged EXEC level and CONFIG levels of the CLI

NOTE: You cannot authenticate IronView Network Manager (SNMP) access to a Foundry device using TACACS/ TACACS+.

The TACACS and TACACS+ protocols define how authentication, authorization, and accounting information is sent between a Foundry device and an authentication database on a TACACS/TACACS+ server. TACACS/TACACS+ services are maintained in a database, typically on a UNIX workstation or PC with a TACACS/TACACS+ server running.

How TACACS+ Differs from TACACS

TACACS is a simple UDP-based access control protocol originally developed by BBN for MILNET. TACACS+ is an enhancement to TACACS and uses TCP to ensure reliable delivery.

TACACS+ is an enhancement to the TACACS security protocol. TACACS+ improves on TACACS by separating the functions of authentication, authorization, and accounting (AAA) and by encrypting all traffic between the Foundry device and the TACACS+ server. TACACS+ allows for arbitrary length and content authentication exchanges, which allow any authentication mechanism to be utilized with the Foundry device. TACACS+ is extensible to provide for site customization and future development features. The protocol allows the Foundry device to request very precise access control and allows the TACACS+ server to respond to each component of that request.

NOTE: TACACS+ provides for authentication, authorization, and accounting, but an implementation or configuration is not required to employ all three.

TACACS/TACACS+ Authentication, Authorization, and Accounting

When you configure a Foundry device to use a TACACS/TACACS+ server for authentication, the device prompts users who are trying to access the CLI for a user name and password, then verifies the password with the TACACS/TACACS+ server.

If you are using TACACS+, Foundry recommends that you also configure ***authorization***, in which the Foundry device consults a TACACS+ server to determine which management privilege level (and which associated set of commands) an authenticated user is allowed to use. You can also optionally configure ***accounting***, which causes the Foundry device to log information on the TACACS+ server when specified events occur on the device.

NOTE: By default, a user logging into the device via Telnet or SSH would first enter the User EXEC level. The user can enter the **enable** command to get to the Privileged EXEC level.

A user that is successfully authenticated can be automatically placed at the Privileged EXEC level after login. See “Entering Privileged EXEC Mode After a Telnet or SSH Login” on page 39-36.

TACACS Authentication

NOTE: Also, multiple challenges are supported for TACACS+ login authentication.

When TACACS authentication takes place, the following events occur:

1. A user attempts to gain access to the Foundry device by doing one of the following:
    - Logging into the device using Telnet, SSH, or the Web management interface
    - Entering the Privileged EXEC level or CONFIG level of the CLI
2. The user is prompted for a username and password.
3. The user enters a username and password.
4. The Foundry device sends a request containing the username and password to the TACACS server.
5. The username and password are validated in the TACACS server’s database.
6. If the password is valid, the user is authenticated.

TACACS+ Authentication

When TACACS+ authentication takes place, the following events occur:

1. A user attempts to gain access to the Foundry device by doing one of the following:
   - Logging into the device using Telnet, SSH, or the Web management interface
   - Entering the Privileged EXEC level or CONFIG level of the CLI
2. The user is prompted for a username.
3. The user enters a username.
4. The Foundry device obtains a password prompt from a TACACS+ server.
5. The user is prompted for a password.
6. The user enters a password.
7. The Foundry device sends the password to the TACACS+ server.
8. The password is validated in the TACACS+ server's database.
9. If the password is valid, the user is authenticated.

TACACS+ Authorization

Foundry devices support two kinds of TACACS+ authorization:

- Exec authorization determines a user's privilege level when they are authenticated
- Command authorization consults a TACACS+ server to get authorization for commands entered by the user

When TACACS+ exec authorization takes place, the following events occur:

1. A user logs into the Foundry device using Telnet, SSH, or the Web management interface
2. The user is authenticated.
3. The Foundry device consults the TACACS+ server to determine the privilege level of the user.
4. The TACACS+ server sends back a response containing an A-V (Attribute-Value) pair with the privilege level of the user.
5. The user is granted the specified privilege level.

When TACACS+ command authorization takes place, the following events occur:

1. A Telnet, SSH, or Web management interface user previously authenticated by a TACACS+ server enters a command on the Foundry device.
2. The Foundry device looks at its configuration to see if the command is at a privilege level that requires TACACS+ command authorization.
3. If the command belongs to a privilege level that requires authorization, the Foundry device consults the TACACS+ server to see if the user is authorized to use the command.
4. If the user is authorized to use the command, the command is executed.

TACACS+ Accounting

TACACS+ accounting works as follows:

1. One of the following events occur on the Foundry device:
   - A user logs into the management interface using Telnet or SSH
   - A user enters a command for which accounting has been configured
   - A system event occurs, such as a reboot or reloading of the configuration file
2. The Foundry device checks the configuration to see if the event is one for which TACACS+ accounting is required.

3. If the event requires TACACS+ accounting, the Foundry device sends a TACACS+ Accounting Start packet to the TACACS+ accounting server, containing information about the event.
4. The TACACS+ accounting server acknowledges the Accounting Start packet.
5. The TACACS+ accounting server records information about the event.
6. When the event is concluded, the Foundry device sends an Accounting Stop packet to the TACACS+ accounting server.
7. The TACACS+ accounting server acknowledges the Accounting Stop packet.

AAA Operations for TACACS/TACACS+

The following table lists the sequence of authentication, authorization, and accounting operations that take place when a user gains access to a Foundry device that has TACACS/TACACS+ security configured.

User Action	Applicable AAA Operations
User attempts to gain access to the Privileged EXEC and CONFIG levels of the CLI	Enable authentication: aaa authentication enable default <method-list>
	Exec authorization (TACACS+): aaa authorization exec default tacacs+
	System accounting start (TACACS+): aaa accounting system default start-stop <method-list>
User logs in using Telnet/SSH	Login authentication: aaa authentication login default <method-list>
	Exec authorization (TACACS+): aaa authorization exec default tacacs+

	Exec accounting start (TACACS+):
	aaa accounting exec default <method-list>
	System accounting start (TACACS+):
	aaa accounting system default start-stop <method-list>

User logs into the Web management interface	Web authentication: aaa authentication web-server default <method-list>
	Exec authorization (TACACS+): aaa authorization exec default tacacs+

User logs out of Telnet/SSH session	Command accounting (TACACS+):
	aaa accounting commands <privilege-level> default start-stop <method-list>
	EXEC accounting stop (TACACS+):
	aaa accounting exec default start-stop <method-list>

User Action	Applicable AAA Operations
User enters system commands (for example, **reload**, **boot system**)	Command authorization (TACACS+): aaa authorization commands <privilege-level> default <method-list>

	Command accounting (TACACS+):
	aaa accounting commands <privilege-level> default start-stop <method-list>
	System accounting stop (TACACS+):
	aaa accounting system default start-stop <method-list>

User enters the command: [no] aaa accounting system default start-stop <method-list>	Command authorization (TACACS+): aaa authorization commands <privilege-level> default <method-list>

	Command accounting (TACACS+):
	aaa accounting commands <privilege-level> default start-stop <method-list>
	System accounting start (TACACS+):
	aaa accounting system default start-stop <method-list>

User enters other commands	Command authorization (TACACS+): aaa authorization commands <privilege-level> default <method-list>
	Command accounting (TACACS+): aaa accounting commands <privilege-level> default start-stop <method-list>

AAA Security for Commands Pasted Into the Running-Config

If AAA security is enabled on the device, commands pasted into the running-config are subject to the same AAA operations as if they were entered manually.

When you paste commands into the running-config, and AAA command authorization and/or accounting is configured on the device, AAA operations are performed on the pasted commands. The AAA operations are performed before the commands are actually added to the running-config. The server performing the AAA operations should be reachable when you paste the commands into the running-config file. If the device determines that a pasted command is invalid, AAA operations are halted on the remaining commands. The remaining commands may not be executed if command authorization is configured.

TACACS/TACACS+ Configuration Considerations

- You must deploy at least one TACACS/TACACS+ server in your network.
- Foundry devices support authentication using up to eight TACACS/TACACS+ servers. The device tries to use the servers in the order you add them to the device's configuration.
- You can select only one primary authentication method for each type of access to a device (CLI through Telnet, CLI Privileged EXEC and CONFIG levels). For example, you can select TACACS+ as the primary authentication method for Telnet CLI access, but you cannot also select RADIUS authentication as a primary method for the same type of access. However, you can configure backup authentication methods for each access type.
- You can configure the Foundry device to authenticate using a TACACS or TACACS+ server, not both.

TACACS Configuration Procedure

For TACACS configurations, use the following procedure:

1. Identify TACACS servers. See “Identifying the TACACS/TACACS+ Servers” on page 39-32.
2. Set optional parameters. See “Setting Optional TACACS/TACACS+ Parameters” on page 39-33.
3. Configure authentication-method lists. See “Configuring Authentication-Method Lists for TACACS/TACACS+” on page 39-35.

TACACS+ Configuration Procedure

For TACACS+ configurations, use the following procedure:

1. Identify TACACS+ servers. See “Identifying the TACACS/TACACS+ Servers” on page 39-32.
2. Set optional parameters. See “Setting Optional TACACS/TACACS+ Parameters” on page 39-33.
3. Configure authentication-method lists. See “Configuring Authentication-Method Lists for TACACS/TACACS+” on page 39-35.
4. Optionally configure TACACS+ authorization. See “Configuring TACACS+ Authorization” on page 39-37.
5. Optionally configure TACACS+ accounting. See “Configuring TACACS+ Accounting” on page 39-39.

Enabling TACACS

TACACS is disabled by default. To configure TACACS/TACACS+ authentication parameters, you must enable TACACS by entering the following command:

```
FastIron(config)# enable snmp config-tacacs
```

Syntax: [no] enable snmp <config-radius | config-tacacs>

The <config-radius> parameter specifies the RADIUS configuration mode. RADIUS is disabled by default.

The <config-tacacs> parameter specifies the TACACS configuration mode. TACACS is disabled by default.

Identifying the TACACS/TACACS+ Servers

To use TACACS/TACACS+ servers to authenticate access to a Foundry device, you must identify the servers to the Foundry device.

For example, to identify three TACACS/TACACS+ servers, enter commands such as the following:

```
FastIron(config)#tacacs-server host 207.94.6.161
FastIron(config)#tacacs-server host 207.94.6.191
FastIron(config)#tacacs-server host 207.94.6.122
```

Syntax: tacacs-server host <ip-addr> | <ipv6-addr> | <hostname> [auth-port <number>]

The <ip-addr>|<ipv6-addr>|<hostname> parameter specifies the IP address or host name of the server. You can enter up to eight **tacacs-server host** commands to specify up to eight different servers.

NOTE: To specify the server's host name instead of its IP address, you must first identify a DNS server using the **ip dns server-address** <ip-addr> command at the global CONFIG level.

If you add multiple TACACS/TACACS+ authentication servers to the Foundry device, the device tries to reach them in the order you add them. For example, if you add three servers in the following order, the software tries the servers in the same order:

1. 207.94.6.161
2. 207.94.6.191
3. 207.94.6.122

You can remove a TACACS/TACACS+ server by entering **no** followed by the **tacacs-server** command. For example, to remove 207.94.6.161, enter the following command:

```
FastIron(config)#no tacacs-server host 207.94.6.161
```

NOTE: If you erase a **tacacs-server** command (by entering "**no**" followed by the command), make sure you also erase the **aaa** commands that specify TACACS/TACACS+ as an authentication method. (See "Configuring Authentication-Method Lists for TACACS/TACACS+" on page 39-35.) Otherwise, when you exit from the CONFIG mode or from a Telnet session, the system continues to believe it is TACACS/TACACS+ enabled and you will not be able to access the system.

The **auth-port** parameter specifies the UDP (for TACACS) or TCP (for TACACS+) port number of the authentication port on the server. The default port number is 49.

Specifying Different Servers for Individual AAA Functions

In a TACACS+ configuration, you can designate a server to handle a specific AAA task. For example, you can designate one TACACS+ server to handle authorization and another TACACS+ server to handle accounting. You can set the TACACS+ key for each server.

To specify different TACACS+ servers for authentication, authorization, and accounting:

```
FastIron(config)#tacacs-server host 1.2.3.4 auth-port 49 authentication-only key
abc
FastIron(config)#tacacs-server host 1.2.3.5 auth-port 49 authorization-only key def
FastIron(config)#tacacs-server host 1.2.3.6 auth-port 49 accounting-only key ghi
```

Syntax: tacacs-server host <ip-addr> | <ipv6-addr> | <server-name> [auth-port <num>] [authentication-only | authorization-only | accounting-only | default] [key 0 | 1 <string>]

The default parameter causes the server to be used for all AAA functions.

After authentication takes place, the server that performed the authentication is used for authorization and/or accounting. If the authenticating server cannot perform the requested function, then the next server in the configured list of servers is tried; this process repeats until a server that can perform the requested function is found, or every server in the configured list has been tried.

Setting Optional TACACS/TACACS+ Parameters

You can set the following optional parameters in a TACACS/TACACS+ configuration:

- TACACS+ key – This parameter specifies the value that the Foundry device sends to the TACACS+ server when trying to authenticate user access.
- Retransmit interval – This parameter specifies how many times the Foundry device will resend an authentication request when the TACACS/TACACS+ server does not respond. The retransmit value can be from 1 – 5 times. The default is 3 times.
- Dead time – This parameter specifies how long the Foundry device waits for the primary authentication server to reply before deciding the server is dead and trying to authenticate using the next server. The dead-time value can be from 1 – 5 seconds. The default is 3 seconds.
- Timeout – This parameter specifies how many seconds the Foundry device waits for a response from a TACACS/TACACS+ server before either retrying the authentication request, or determining that the TACACS/TACACS+ servers are unavailable and moving on to the next authentication method in the authentication-method list. The timeout can be from 1 – 15 seconds. The default is 3 seconds.
- TACACS/TACACS+ over IPv6 – This parameter enables the Foundry device to send TACACS/TACACS+ packets over IPv6.

Setting the TACACS+ Key

The **key** parameter in the **tacacs-server** command is used to encrypt TACACS+ packets before they are sent over the network. The value for the **key** parameter on the Foundry device should match the one configured on the TACACS+ server. The key can be from 1 – 32 characters in length and cannot include any space characters.

NOTE: The **tacacs-server key** command applies only to TACACS+ servers, not to TACACS servers. If you are configuring TACACS, do not configure a key on the TACACS server and do not enter a key on the Foundry device.

To specify a TACACS+ server key:

```
FastIron(config)#tacacs-server key rkwong
```

Syntax: tacacs-server key [0 | 1] <string>

When you display the configuration of the Foundry device, the TACACS+ keys are encrypted. For example:

```
FastIron(config)#tacacs-server key 1 abc
FastIron(config)#write terminal
...
tacacs-server host 1.2.3.5 auth-port 49
tacacs key 1 $!2d
```

NOTE: Encryption of the TACACS+ keys is done by default. The **0** parameter disables encryption. The **1** parameter is not required; it is provided for backwards compatibility.

Setting the Retransmission Limit

The **retransmit** parameter specifies how many times the Foundry device will resend an authentication request when the TACACS/TACACS+ server does not respond. The retransmit limit can be from 1 – 5 times. The default is 3 times.

To set the TACACS/TACACS+ retransmit limit:

```
FastIron(config)#tacacs-server retransmit 5
```

Syntax: tacacs-server retransmit <number>

Setting the Timeout Parameter

The **timeout** parameter specifies how many seconds the Foundry device waits for a response from the TACACS/ TACACS+ server before either retrying the authentication request, or determining that the TACACS/TACACS+ server is unavailable and moving on to the next authentication method in the authentication-method list. The timeout can be from 1 – 15 seconds. The default is 3 seconds.

```
FastIron(config)#tacacs-server timeout 5
```

Syntax: tacacs-server timeout <number>

TACACS/TACACS+ over IPv6

Platform Support:

- FESX and FSX devices running software release 02.5.00 and later

To enable this feature, enter a command such as the following at the Global CONFIG level of the CLI:

```
FastIron(config)#tacacs-server host ipv6 3000::200
```

Syntax: tacacs-server host ipv6 <ipv6-host address>

The <ipv6-host address> is the IPv6 address of the TACACS server. When you enter the IPv6 host address, you do not need to specify the prefix length. A prefix length of 128 is implied.

Configuring Authentication-Method Lists for TACACS/TACACS+

You can use TACACS/TACACS+ to authenticate Telnet/SSH access and access to Privileged EXEC level and CONFIG levels of the CLI. When configuring TACACS/TACACS+ authentication, you create authentication-method lists specifically for these access methods, specifying TACACS/TACACS+ as the primary authentication method.

Within the authentication-method list, TACACS/TACACS+ is specified as the primary authentication method and up to six backup authentication methods are specified as alternates. If TACACS/TACACS+ authentication fails due to an error, the device tries the backup authentication methods in the order they appear in the list.

When you configure authentication-method lists for TACACS/TACACS+ authentication, you must create a separate authentication-method list for Telnet/SSH CLI access, and for access to the Privileged EXEC level and CONFIG levels of the CLI.

To create an authentication method list that specifies TACACS/TACACS+ as the primary authentication method for securing Telnet/SSH access to the CLI:

```
FastIron(config)#enable telnet authentication
FastIron(config)#aaa authentication login default tacacs local
```

The commands above cause TACACS/TACACS+ to be the primary authentication method for securing Telnet/SSH access to the CLI. If TACACS/TACACS+ authentication fails due to an error with the server, authentication is performed using local user accounts instead.

To create an authentication-method list that specifies TACACS/TACACS+ as the primary authentication method for securing access to Privileged EXEC level and CONFIG levels of the CLI:

```
FastIron(config)#aaa authentication enable default tacacs local none
```

The command above causes TACACS/TACACS+ to be the primary authentication method for securing access to Privileged EXEC level and CONFIG levels of the CLI. If TACACS/TACACS+ authentication fails due to an error with the server, local authentication is used instead. If local authentication fails, no authentication is used; the device automatically permits access.

Syntax: [no] aaa authentication enable | login default <method1> [<method2>] [<method3>] [<method4>] [<method5>] [<method6>] [<method7>]

The **web-server | enable | login** parameter specifies the type of access this authentication-method list controls. You can configure one authentication-method list for each type of access.

NOTE: If you configure authentication for Web management access, authentication is performed each time a page is requested from the server. When frames are enabled on the Web management interface, the browser sends an HTTP request for each frame. The Foundry device authenticates each HTTP request from the browser. To limit authentications to one per page, disable frames on the Web management interface.

The <method1> parameter specifies the primary authentication method. The remaining optional <method> parameters specify additional methods to try if an error occurs with the primary method. A method can be one of the values listed in the Method Parameter column in the following table.

Table 39.2: Authentication Method Values

Method Parameter	Description
line	Authenticate using the password you configured for Telnet access. The Telnet password is configured using the **enable telnet password…** command. See “Setting a Telnet Password” on page 39-13.
enable	Authenticate using the password you configured for the Super User privilege level. This password is configured using the **enable super-user-password…** command. See “Setting Passwords for Management Privilege Levels” on page 39-13.

Table 39.2: Authentication Method Values (Continued)

Method Parameter	Description
local	Authenticate using a local user name and password you configured on the device. Local user names and passwords are configured using the **username…** command. See “Configuring a Local User Account” on page 39-21.
tacacs	Authenticate using the database on a TACACS server. You also must identify the server to the device using the **tacacs-server** command.
tacacs+	Authenticate using the database on a TACACS+ server. You also must identify the server to the device using the **tacacs-server** command.
radius	Authenticate using the database on a RADIUS server. You also must identify the server to the device using the **radius-server** command.
none	Do not use any authentication method. The device automatically permits access.

NOTE: For examples of how to define authentication-method lists for types of authentication other than TACACS/TACACS+, see “Configuring Authentication-Method Lists” on page 39-57.

Entering Privileged EXEC Mode After a Telnet or SSH Login

By default, a user enters User EXEC mode after a successful login through Telnet or SSH. Optionally, you can configure the device so that a user enters Privileged EXEC mode after a Telnet or SSH login. To do this, use the following command:

```
FastIron(config)#aaa authentication login privilege-mode
```

Syntax: aaa authentication login privilege-mode

The user’s privilege level is based on the privilege level granted during login.

Configuring Enable Authentication to Prompt for Password Only

If Enable authentication is configured on the device, when a user attempts to gain Super User access to the Privileged EXEC and CONFIG levels of the CLI, by default he or she is prompted for a username and password. You can configure the Foundry device to prompt only for a password. The device uses the username entered at login, if one is available. If no username was entered at login, the device prompts for both username and password.

To configure the Foundry device to prompt only for a password when a user attempts to gain Super User access to the Privileged EXEC and CONFIG levels of the CLI:

```
FastIron(config)#aaa authentication enable implicit-user
```

Syntax: [no] aaa authentication enable implicit-user

Telnet/SSH Prompts When the TACACS+ Server is Unavailable

When TACACS+ is the first method in the authentication method list, the device displays the login prompt received from the TACACS+ server. If a user attempts to login through Telnet or SSH, but none of the configured TACACS+ servers are available, the following takes place:

- If the next method in the authentication method list is "enable", the login prompt is skipped, and the user is prompted for the Enable password (that is, the password configured with the **enable super-user-password** command).
- If the next method in the authentication method list is "line", the login prompt is skipped, and the user is prompted for the Line password (that is, the password configured with the **enable telnet password** command).

Configuring TACACS+ Authorization

Foundry devices support TACACS+ authorization for controlling access to management functions in the CLI. Two kinds of TACACS+ authorization are supported:

- Exec authorization determines a user's privilege level when they are authenticated
- Command authorization consults a TACACS+ server to get authorization for commands entered by the user

Configuring Exec Authorization

When TACACS+ exec authorization is performed, the Foundry device consults a TACACS+ server to determine the privilege level of the authenticated user. To configure TACACS+ exec authorization on the Foundry device, enter the following command:

```
FastIron(config)#aaa authorization exec default tacacs+
```

Syntax: aaa authorization exec default tacacs+ | none

If you specify **none**, or omit the **aaa authorization exec** command from the device's configuration, no exec authorization is performed.

A user's privilege level is obtained from the TACACS+ server in the "foundry-privlvl" A-V pair. If the **aaa authorization exec default tacacs** command exists in the configuration, the device assigns the user the privilege level specified by this A-V pair. If the command does not exist in the configuration, then the value in the "foundry-privlvl" A-V pair is ignored, and the user is granted Super User access.

NOTE: If the **aaa authorization exec default tacacs+** command exists in the configuration, following successful authentication the device assigns the user the privilege level specified by the "foundry-privlvl" A-V pair received from the TACACS+ server. If the **aaa authorization exec default tacacs+** command does not exist in the configuration, then the value in the "foundry-privlvl" A-V pair is ignored, and the user is granted Super User access.

Also note that in order for the **aaa authorization exec default tacacs+** command to work, either the **aaa authentication enable default tacacs+** command, or the **aaa authentication login privilege-mode** command must also exist in the configuration.

Configuring an Attribute-Value Pair on the TACACS+ Server

During TACACS+ exec authorization, the Foundry device expects the TACACS+ server to send a response containing an A-V (Attribute-Value) pair that specifies the privilege level of the user. When the Foundry device receives the response, it extracts an A-V pair configured for the Exec service and uses it to determine the user's privilege level.

To set a user's privilege level, you can configure the "foundry-privlvl" A-V pair for the Exec service on the TACACS+ server. For example:

```
user=bob {
   default service = permit
   member admin
   #Global password
   global = cleartext "cat"
   service = exec {
     foundry-privlvl = 0
        }
}
```

In this example, the A-V pair `foundry-privlvl = 0` grants the user full read-write access. The value in the foundry-privlvl A-V pair is an integer that indicates the privilege level of the user. Possible values are 0 for super-user level, 4 for port-config level, or 5 for read-only level. If a value other than 0, 4, or 5 is specified in the foundry-privlvl A-V pair, the default privilege level of 5 (read-only) is used. The foundry-privlvl A-V pair can also be embedded in the group configuration for the user. See your TACACS+ documentation for the configuration syntax relevant to your server.

If the foundry-privlvl A-V pair is not present, the Foundry device extracts the last A-V pair configured for the Exec service that has a numeric value. The Foundry device uses this A-V pair to determine the user's privilege level. For example:

```
user=bob {
   default service = permit
   member admin
   #Global password
   global = cleartext "cat"
   service = exec {
     privlvl = 15
        }
}
```

The attribute name in the A-V pair is not significant; the Foundry device uses the last one that has a numeric value. However, the Foundry device interprets the value for a non-"foundry-privlvl" A-V pair differently than it does for a "foundry-privlvl" A-V pair. The following table lists how the Foundry device associates a value from a non-"foundry-privlvl" A-V pair with a Foundry privilege level.

Table 39.3: Foundry Equivalents for non-"foundry-privlvl" A-V Pair Values

Value for non-"foundry-privlvl" A-V Pair	Foundry Privilege Level
15	0 (super-user)
From 14 – 1	4 (port-config)
Any other number or 0	5 (read-only)

In the example above, the A-V pair configured for the Exec service is `privlvl = 15`. The Foundry device uses the value in this A-V pair to set the user's privilege level to 0 (super-user), granting the user full read-write access.

In a configuration that has both a "foundry-privlvl" A-V pair and a non-"foundry-privlvl" A-V pair for the Exec service, the non-"foundry-privlvl" A-V pair is ignored. For example:

```
user=bob {
   default service = permit
   member admin
   #Global password
   global = cleartext "cat"
   service = exec {
     foundry-privlvl = 4
     privlvl = 15
        }
}
```

In this example, the user would be granted a privilege level of 4 (port-config level). The `privlvl = 15` A-V pair is ignored by the Foundry device.

If the TACACS+ server has no A-V pair configured for the Exec service, the default privilege level of 5 (read-only) is used.

Configuring Command Authorization

When TACACS+ command authorization is enabled, the Foundry device consults a TACACS+ server to get authorization for commands entered by the user.

You enable TACACS+ command authorization by specifying a privilege level whose commands require authorization. For example, to configure the Foundry device to perform authorization for the commands available at the Super User privilege level (that is, all commands on the device), enter the following command:

```
FastIron(config)#aaa authorization commands 0 default tacacs+
```

Syntax: aaa authorization commands <privilege-level> default tacacs+ | radius | none

The <privilege-level> parameter can be one of the following:

- **0** – Authorization is performed for commands available at the Super User level (all commands)
- **4** – Authorization is performed for commands available at the Port Configuration level (port-config and read-only commands)
- **5** – Authorization is performed for commands available at the Read Only level (read-only commands)

NOTE: TACACS+ command authorization can be performed only for commands entered from Telnet or SSH sessions, or from the console. No authorization is performed for commands entered at the Web management interface or IronView Network Manager.

TACACS+ command authorization is not performed for the following commands:

- At all levels: **exit**, **logout**, **end**, and **quit**.
- At the Privileged EXEC level: **enable** or **enable** <text>, where <text> is the password configured for the Super User privilege level.

If configured, command accounting is performed for these commands.

AAA Support for Console Commands

AAA support for commands entered at the console includes the following:

- Login prompt that uses AAA authentication, using authentication-method Lists[1]
- Exec Authorization[1]
- Exec Accounting[1]
- Command authorization
- Command accounting
- System Accounting[2]

To enable AAA support for commands entered at the console, enter the following command:

```
FastIron(config)#enable aaa console
```

Syntax: [no] enable aaa console

Configuring TACACS+ Accounting

Foundry devices support TACACS+ accounting for recording information about user activity and system events. When you configure TACACS+ accounting on a Foundry device, information is sent to a TACACS+ accounting server when specified events occur, such as when a user logs into the device or the system is rebooted.

Configuring TACACS+ Accounting for Telnet/SSH (Shell) Access

To send an Accounting Start packet to the TACACS+ accounting server when an authenticated user establishes a Telnet or SSH session on the Foundry device, and an Accounting Stop packet when the user logs out:

```
FastIron(config)#aaa accounting exec default start-stop tacacs+
```

Syntax: aaa accounting exec default start-stop radius | tacacs+ | none

Configuring TACACS+ Accounting for CLI Commands

You can configure TACACS+ accounting for CLI commands by specifying a privilege level whose commands require accounting. For example, to configure the Foundry device to perform TACACS+ accounting for the

---

[1] Supported in software releases 02.4.00 and later.
[2] Supported in software releases 02.4.00 and later.

commands available at the Super User privilege level (that is; all commands on the device), enter the following command:

```
FastIron(config)#aaa accounting commands 0 default start-stop tacacs+
```

An Accounting Start packet is sent to the TACACS+ accounting server when a user enters a command, and an Accounting Stop packet is sent when the service provided by the command is completed.

NOTE: If authorization is enabled, and the command requires authorization, then authorization is performed before accounting takes place. If authorization fails for the command, no accounting takes place.

Syntax: aaa accounting commands <privilege-level> default start-stop radius | tacacs+ | none

The <privilege-level> parameter can be one of the following:

- **0** – Records commands available at the Super User level (all commands)
- **4** – Records commands available at the Port Configuration level (port-config and read-only commands)
- **5** – Records commands available at the Read Only level (read-only commands)

Configuring TACACS+ Accounting for System Events

You can configure TACACS+ accounting to record when system events occur on the Foundry device. System events include rebooting and when changes to the active configuration are made.

The following command causes an Accounting Start packet to be sent to the TACACS+ accounting server when a system event occurs, and a Accounting Stop packet to be sent when the system event is completed:

```
FastIron(config)#aaa accounting system default start-stop tacacs+
```

Syntax: aaa accounting system default start-stop radius | tacacs+ | none

Configuring an Interface as the Source for All TACACS/TACACS+ Packets

You can designate the lowest-numbered IP address configured an Ethernet port, loopback interface, or virtual interface as the source IP address for all TACACS/TACACS+ packets from the Layer 3 Switch. Identifying a single source IP address for TACACS/TACACS+ packets provides the following benefits:

- If your TACACS/TACACS+ server is configured to accept packets only from specific links or IP addresses, you can use this feature to simplify configuration of the TACACS/TACACS+ server by configuring the Foundry device to always send the TACACS/TACACS+ packets from the same link or source address.
- If you specify a loopback interface as the single source for TACACS/TACACS+ packets, TACACS/TACACS+ servers can receive the packets regardless of the states of individual links. Thus, if a link to the TACACS/TACACS+ server becomes unavailable but the client or server can be reached through another link, the client or server still receives the packets, and the packets still have the source IP address of the loopback interface.

The software contains separate CLI commands for specifying the source interface for Telnet, TACACS/TACACS+, and RADIUS packets. You can configure a source interface for one or more of these types of packets.

To specify an Ethernet, loopback, or virtual interface as the source for all TACACS/TACACS+ packets from the device, use the following CLI method. The software uses the lowest-numbered IP address configured on the port or interface as the source IP address for TACACS/TACACS+ packets originated by the device.

To specify the lowest-numbered IP address configured on a virtual interface as the device's source for all TACACS/TACACS+ packets, enter commands such as the following:

```
FastIron(config)#int ve 1
FastIron(config-vif-1)#ip address 10.0.0.3/24
FastIron(config-vif-1)#exit
FastIron(config)#ip tacacs source-interface ve 1
```

The commands in this example configure virtual interface 1, assign IP address 10.0.0.3/24 to the interface, then designate the interface as the source for all TACACS/TACACS+ packets from the Layer 3 Switch.

Syntax: ip tacacs source-interface ethernet [<slotnum>/]<portnum> | loopback <num> | ve <num>

The <slotnum> parameter is required on chassis devices.

The <portnum> parameter is a valid port number.

The <num> parameter is a loopback interface or virtual interface number.

Displaying TACACS/TACACS+ Statistics and Configuration Information

The **show aaa** command displays information about all TACACS+ and RADIUS servers identified on the device. For example:

```
FastIron#show aaa
Tacacs+ key: foundry
Tacacs+ retries: 1
Tacacs+ timeout: 15 seconds
Tacacs+ dead-time: 3 minutes
Tacacs+ Server: 207.95.6.90 Port:49:
                opens=6 closes=3 timeouts=3 errors=0
                packets in=4 packets out=4
no connection

Radius key: networks
Radius retries: 3
Radius timeout: 3 seconds
Radius dead-time: 3 minutes
Radius Server:  207.95.6.90 Auth Port=1645 Acct Port=1646:
                opens=2 closes=1 timeouts=1 errors=0
                packets in=1 packets out=4
no connection
```

The following table describes the TACACS/TACACS+ information displayed by the **show aaa** command.

Table 39.4: Output of the show aaa command for TACACS/TACACS+

Field	Description
Tacacs+ key	The setting configured with the **tacacs-server key** command. At the Super User privilege level, the actual text of the key is displayed. At the other privilege levels, a string of periods (....) is displayed instead of the text.
Tacacs+ retries	The setting configured with the **tacacs-server retransmit** command.
Tacacs+ timeout	The setting configured with the **tacacs-server timeout** command.
Tacacs+ dead-time	The setting configured with the **tacacs-server dead-time** command.

Table 39.4: Output of the show aaa command for TACACS/TACACS+

Field	Description	
Tacacs+ Server	For each TACACS/TACACS+ server, the IP address, port, and the following statistics are displayed:	
	opens	Number of times the port was opened for communication with the server
	closes	Number of times the port was closed normally
	timeouts	Number of times port was closed due to a timeout
	errors	Number of times an error occurred while opening the port
	packets in	Number of packets received from the server
	packets out	Number of packets sent to the server
connection	The current connection status. This can be “no connection” or “connection active”.	

The **show web** command displays the privilege level of Web management interface users. For example:

```
FastIron#show web
User                                    Privilege       IP address
set                                             0       192.168.1.234
```

Syntax: show web

Configuring RADIUS Security

You can use a Remote Authentication Dial In User Service (RADIUS) server to secure the following types of access to the Foundry Layer 2 Switch or Layer 3 Switch:

- Telnet access
- SSH access
- Web management access
- Access to the Privileged EXEC level and CONFIG levels of the CLI

NOTE: Foundry devices do not support RADIUS security for SNMP (IronView Network Manager) access.

RADIUS Authentication, Authorization, and Accounting

When RADIUS ***authentication*** is implemented, the Foundry device consults a RADIUS server to verify user names and passwords. You can optionally configure RADIUS ***authorization***, in which the Foundry device consults a list of commands supplied by the RADIUS server to determine whether a user can execute a command he or she has entered, as well as ***accounting***, which causes the Foundry device to log information on a RADIUS accounting server when specified events occur on the device.

RADIUS Authentication

When RADIUS authentication takes place, the following events occur:

1. A user attempts to gain access to the Foundry device by doing one of the following:
    - Logging into the device using Telnet, SSH, or the Web management interface
    - Entering the Privileged EXEC level or CONFIG level of the CLI

2. The user is prompted for a username and password.
3. The user enters a username and password.
4. The Foundry device sends a RADIUS Access-Request packet containing the username and password to the RADIUS server.
5. The RADIUS server validates the Foundry device using a shared secret (the RADIUS key).
6. The RADIUS server looks up the username in its database.
7. If the username is found in the database, the RADIUS server validates the password.
8. If the password is valid, the RADIUS server sends an Access-Accept packet to the Foundry device, authenticating the user. Within the Access-Accept packet are three Foundry vendor-specific attributes that indicate:
    - The privilege level of the user
    - A list of commands
    - Whether the user is allowed or denied usage of the commands in the list

    The last two attributes are used with RADIUS authorization, if configured.
9. The user is authenticated, and the information supplied in the Access-Accept packet for the user is stored on the Foundry device. The user is granted the specified privilege level. If you configure RADIUS authorization, the user is allowed or denied usage of the commands in the list.

RADIUS Authorization

When RADIUS authorization takes place, the following events occur:

1. A user previously authenticated by a RADIUS server enters a command on the Foundry device.
2. The Foundry device looks at its configuration to see if the command is at a privilege level that requires RADIUS command authorization.
3. If the command belongs to a privilege level that requires authorization, the Foundry device looks at the list of commands delivered to it in the RADIUS Access-Accept packet when the user was authenticated. (Along with the command list, an attribute was sent that specifies whether the user is permitted or denied usage of the commands in the list.)

    NOTE: After RADIUS authentication takes place, the command list resides on the Foundry device. The RADIUS server is not consulted again once the user has been authenticated. This means that any changes made to the user’s command list on the RADIUS server are not reflected until the next time the user is authenticated by the RADIUS server, and the new command list is sent to the Foundry device.

4. If the command list indicates that the user is authorized to use the command, the command is executed.

RADIUS Accounting

RADIUS accounting works as follows:

1. One of the following events occur on the Foundry device:
    - A user logs into the management interface using Telnet or SSH
    - A user enters a command for which accounting has been configured
    - A system event occurs, such as a reboot or reloading of the configuration file
2. The Foundry device checks its configuration to see if the event is one for which RADIUS accounting is required.
3. If the event requires RADIUS accounting, the Foundry device sends a RADIUS Accounting Start packet to the RADIUS accounting server, containing information about the event.
4. The RADIUS accounting server acknowledges the Accounting Start packet.

5. The RADIUS accounting server records information about the event.
6. When the event is concluded, the Foundry device sends an Accounting Stop packet to the RADIUS accounting server.
7. The RADIUS accounting server acknowledges the Accounting Stop packet.

AAA Operations for RADIUS

The following table lists the sequence of authentication, authorization, and accounting operations that take place when a user gains access to a Foundry device that has RADIUS security configured.

User Action	Applicable AAA Operations
User attempts to gain access to the Privileged EXEC and CONFIG levels of the CLI	Enable authentication: aaa authentication enable default <method-list>
	System accounting start: aaa accounting system default start-stop <method-list>
User logs in using Telnet/SSH	Login authentication: aaa authentication login default <method-list>
	EXEC accounting Start: aaa accounting exec default start-stop <method-list> System accounting Start: aaa accounting system default start-stop <method-list>
User logs into the Web management interface	Web authentication: aaa authentication web-server default <method-list>
User logs out of Telnet/SSH session	Command authorization for **logout** command: aaa authorization commands <privilege-level> default <method-list>
	Command accounting: aaa accounting commands <privilege-level> default start-stop <method-list> EXEC accounting stop: aaa accounting exec default start-stop <method-list>
User enters system commands (for example, **reload**, **boot system**)	Command authorization: aaa authorization commands <privilege-level> default <method-list>
	Command accounting: aaa accounting commands <privilege-level> default start-stop <method-list> System accounting stop: aaa accounting system default start-stop <method-list>

User Action	Applicable AAA Operations
User enters the command: [no] aaa accounting system default start-stop <method-list>	Command authorization: aaa authorization commands <privilege-level> default <method-list>
	Command accounting: aaa accounting commands <privilege-level> default start-stop <method-list> System accounting start: aaa accounting system default start-stop <method-list>
User enters other commands	Command authorization: aaa authorization commands <privilege-level> default <method-list>
	Command accounting: aaa accounting commands <privilege-level> default start-stop <method-list>

AAA Security for Commands Pasted Into the Running-Config

If AAA security is enabled on the device, commands pasted into the running-config are subject to the same AAA operations as if they were entered manually.

When you paste commands into the running-config, and AAA command authorization and/or accounting is configured on the device, AAA operations are performed on the pasted commands. The AAA operations are performed before the commands are actually added to the running-config. The server performing the AAA operations should be reachable when you paste the commands into the running-config file. If the device determines that a pasted command is invalid, AAA operations are halted on the remaining commands. The remaining commands may not be executed if command authorization is configured.

NOTE: Since RADIUS command authorization relies on a list of commands received from the RADIUS server when authentication is performed, it is important that you use RADIUS authentication when you also use RADIUS command authorization.

RADIUS Configuration Considerations

- You must deploy at least one RADIUS server in your network.
- Foundry devices support authentication using up to eight RADIUS servers. The device tries to use the servers in the order you add them to the device's configuration. If one RADIUS server is not responding, the Foundry device tries the next one in the list.
- Starting in software release FSX 04.0.00, you can optionally configure a RADIUS server as a ***port server***, indicating that the server will be used only to authenticate users on ports to which it is mapped, as opposed to globally authenticating users on all ports of the device. In previous releases, all configured RADIUS servers are "global" servers and apply to users on all ports of the device. See "Configuring a RADIUS Server per Port" on page 39-48.
- Starting in software release FSX 04.0.00, you can map up to eight RADIUS servers to each port on the Foundry device. The port will authenticate users using only the RADIUS servers to which it is mapped. If there are no RADIUS servers mapped to a port, it will use the "global" servers for authentication. In previous releases, all RADIUS servers are "global" servers and cannot be bound to individual ports. See "Mapping a RADIUS Server to Individual Ports" on page 39-49.

- You can select only one primary authentication method for each type of access to a device (CLI through Telnet, CLI Privileged EXEC and CONFIG levels). For example, you can select RADIUS as the primary authentication method for Telnet CLI access, but you cannot also select TACACS+ authentication as the primary method for the same type of access. However, you can configure backup authentication methods for each access type.

RADIUS Configuration Procedure

Use the following procedure to configure a Foundry device for RADIUS:

1. Configure Foundry vendor-specific attributes on the RADIUS server. See “Configuring Foundry-Specific Attributes on the RADIUS Server” on page 39-46.
2. Identify the RADIUS server to the Foundry device. See “Identifying the RADIUS Server to the Foundry Device” on page 39-48.
3. Optionally specify different servers for individual AAA functions. See “Specifying Different Servers for Individual AAA Functions” on page 39-48.
4. Optionally configure the RADIUS server as a “port only” server. See “Configuring a RADIUS Server per Port” on page 39-48.
5. Optionally bind the RADIUS server(s) to ports on the Foundry device. See “Mapping a RADIUS Server to Individual Ports” on page 39-49.
6. Set RADIUS parameters. See “Setting RADIUS Parameters” on page 39-50.
7. Configure authentication-method lists. See “Configuring Authentication-Method Lists for RADIUS” on page 39-51.
8. Optionally configure RADIUS authorization. See “Configuring RADIUS Authorization” on page 39-53.
9. Optionally configure RADIUS accounting. “Configuring RADIUS Accounting” on page 39-54.

Configuring Foundry-Specific Attributes on the RADIUS Server

NOTE: For all Foundry devices, RADIUS Challenge is supported for 802.1x authentication but not for login authentication.

During the RADIUS authentication process, if a user supplies a valid username and password, the RADIUS server sends an Access-Accept packet to the Foundry device, authenticating the user. Within the Access-Accept packet are three Foundry vendor-specific attributes that indicate:

- The privilege level of the user
- A list of commands
- Whether the user is allowed or denied usage of the commands in the list

You must add these three Foundry vendor-specific attributes to your RADIUS server’s configuration, and configure the attributes in the individual or group profiles of the users that will access the Foundry device.

Foundry's Vendor-ID is 1991, with Vendor-Type 1. The following table describes the Foundry vendor-specific attributes.

Table 39.5: Foundry vendor-specific attributes for RADIUS

Attribute Name	Attribute ID	Data Type	Description
foundry-privilege-level	1	integer	Specifies the privilege level for the user. This attribute can be set to one of the following: **0** Super User level – Allows complete read-and-write access to the system. This is generally for system administrators and is the only management privilege level that allows you to configure passwords. **4** Port Configuration level – Allows read-and-write access for specific ports but not for global (system-wide) parameters. **5** Read Only level – Allows access to the Privileged EXEC mode and CONFIG mode of the CLI but only with read access.
foundry-command-string	2	string	Specifies a list of CLI commands that are permitted or denied to the user when RADIUS authorization is configured. The commands are delimited by semi-colons (;). You can specify an asterisk (*) as a wildcard at the end of a command string. For example, the following command list specifies all **show** and **debug ip** commands, as well as the **write terminal** command: show *; debug ip *; write term*
foundry-command-exception-flag	3	integer	Specifies whether the commands indicated by the foundry-command-string attribute are permitted or denied to the user. This attribute can be set to one of the following: **0** Permit execution of the commands indicated by foundry-command-string, deny all other commands. **1** Deny execution of the commands indicated by foundry-command-string, permit all other commands.

Enabling SNMP to Configure RADIUS

To enable SNMP access to RADIUS MIB objects on the device, enter a command such as the following:

```
FastIron(config)#enable snmp config-radius
```

Syntax: [no] enable snmp <config-radius | config-tacacs>

The <config-radius> parameter specifies the RADIUS configuration mode. RADIUS is disabled by default.

The <config-tacacs> parameter specifies the TACACS configuration mode. TACACS is disabled by default.

Identifying the RADIUS Server to the Foundry Device

To use a RADIUS server to authenticate access to a Foundry device, you must identify the server to the Foundry device. For example:

```
FastIron(config)#radius-server host 209.157.22.99
```

Syntax: radius-server host <ip-addr> | <iipv6-addr> | <server-name> [auth-port <number>] [acct-port <number>]

The **host** <ip-addr> | <ipv6-addr> | <server-name> parameter is either an IP address or an ASCII text string.

The <auth-port> parameter is the Authentication port number; it is an optional parameter. The default is 1645.

The <acct-port> parameter is the Accounting port number; it is an optional parameter. The default is 1646.

Specifying Different Servers for Individual AAA Functions

In a RADIUS configuration, you can designate a server to handle a specific AAA task. For example, you can designate one RADIUS server to handle authorization and another RADIUS server to handle accounting. You can specify individual servers for authentication and accounting, but not for authorization. You can set the RADIUS key for each server.

To specify different RADIUS servers for authentication, authorization, and accounting:

```
FastIron(config)#radius-server host 1.2.3.4 authentication-only key abc
FastIron(config)#radius-server host 1.2.3.5 authorization-only key def
FastIron(config)#radius-server host 1.2.3.6 accounting-only key ghi
```

Syntax: radius-server host <ip-addr> | <ipv6-addr> | <server-name> [auth-port <number>] [acct-port <number>] [authentication-only | accounting-only | default] [key 0 | 1 <string>]

The **default** parameter causes the server to be used for all AAA functions.

After authentication takes place, the server that performed the authentication is used for authorization and/or accounting. If the authenticating server cannot perform the requested function, then the next server in the configured list of servers is tried; this process repeats until a server that can perform the requested function is found, or every server in the configured list has been tried.

Configuring a RADIUS Server per Port

Platform Support:

- FESX/FSX/FWSX devices running software release 04.0.00 and later – L2, BL3, L3

You can optionally configure a RADIUS server per port, indicating that it will be used only to authenticate users on ports to which it is mapped. A RADIUS server that is not explicitly configured as a RADIUS server per port is a ***global server***, and can be used to authenticate users on ports to which no RADIUS servers are mapped.

Configuration Notes

- This feature works with 802.1X and multi-device port authentication only.
- As in previous releases, you can define up to eight RADIUS servers per Foundry device.

Configuration Example and Command Syntax

The following shows an example configuration:

```
FastIron(config)#radius-server host 10.10.10.103 auth-port 1812 acct-port 1813
default key mykeyword dot1x port-only
FastIron(config)#radius-server host 10.10.10.104 auth-port 1812 acct-port 1813
default key mykeyword dot1x port-only
FastIron(config)#radius-server host 10.10.10.105 auth-port 1812 acct-port 1813
default key mykeyword dot1x
FastIron(config)#radius-server host 10.10.10.106 auth-port 1812 acct-port 1813
default key mykeyword dot1x
```

The above configuration has the following affect:

- RADIUS servers 10.10.10.103 and 10.10.10.104 will be used only to authenticate users on ports to which the servers are mapped. To map a RADIUS server to a port, see “Mapping a RADIUS Server to Individual Ports” on page 39-49.
- RADIUS servers 10.10.10.105 and 10.10.10.106 will be used to authenticate users on ports to which no RADIUS servers are mapped. For example, port e 9, to which no RADIUS servers are mapped, will send a RADIUS request to the first configured RADIUS server, 10.10.10.105. If the request fails, it will go to the second configured RADIUS server, 10.10.10.106. It will not send requests to 10.10.10.103 or 10.10.10.104, since these servers are configured as port servers.

Syntax: radius-server host <ip-addr> | <server-name> [auth-port <number>] [acct-port <number>] [default key <string> dot1x] [port-only]

The **host** <ip-addr> is the IPv4 address.

The **auth-port** <number> parameter is the Authentication port number; it is an optional parameter. The default is 1645.

The **acct-port** <number> parameter is the Accounting port number; it is an optional parameter. The default is 1646.

The **default key** <string> **dot1x** parameter indicates that this RADIUS server supports the 802.1X standard. A RADIUS server that supports the 802.1X standard can also be used to authenticate non-802.1X authentication requests.

The **port-only** parameter is optional and specifies that the server will be used only to authenticate users on ports to which it is mapped.

Mapping a RADIUS Server to Individual Ports

Platform Support:

- FESX/FSX/FWSX devices running software release 04.0.00 and later – L2, BL3, L3

You can map up to eight RADIUS servers to each port on the Foundry device. The port will authenticate users using only the RADIUS servers to which the port is mapped. If there are no RADIUS servers mapped to a port, it will use the “global” servers for authentication.

As in previous releases, a port goes through the list of servers in the order in which it was mapped or configured, until a server that can perform the requested function is found, or until every server in the list has been tried.

Configuration Notes

- This feature works with 802.1X and multic-device port authentication only.
- You can map a RADIUS server to a physical port only. You cannot map a RADIUS server to a VE.

Configuration Example and Command Syntax

To map a RADIUS server to a port, enter commands such as the following:

```
FastIron(config)#int e 3
FastIron(config-if-e1000-3)#dot1x port-control auto
FastIron(config-if-e1000-3)#use-radius-server 10.10.10.103
FastIron(config-if-e1000-3)#use-radius-server 10.10.10.110
```

With the above configuration, port e 3 would send a RADIUS request to 10.10.10.103 first, since it is the first server mapped to the port. If it fails, it will go to 10.10.10.110.

Syntax: use-radius-server <ip-addr>

The **host** <ip-addr> is an IPv4 address.

Setting RADIUS Parameters

You can set the following parameters in a RADIUS configuration:

- RADIUS key – This parameter specifies the value that the Foundry device sends to the RADIUS server when trying to authenticate user access.
- Retransmit interval – This parameter specifies how many times the Foundry device will resend an authentication request when the RADIUS server does not respond. The retransmit value can be from 1 – 5 times. The default is 3 times.
- Timeout – This parameter specifies how many seconds the Foundry device waits for a response from a RADIUS server before either retrying the authentication request, or determining that the RADIUS servers are unavailable and moving on to the next authentication method in the authentication-method list. The timeout can be from 1 – 15 seconds. The default is 3 seconds.

Setting the RADIUS Key

The **key** parameter in the **radius-server** command is used to encrypt RADIUS packets before they are sent over the network. The value for the **key** parameter on the Foundry device should match the one configured on the RADIUS server. The key can be from 1 – 32 characters in length and cannot include any space characters.

To specify a RADIUS server key:

```
FastIron(config)#radius-server key mirabeau
```

Syntax: radius-server key [0 | 1] <string>

When you display the configuration of the Foundry device, the RADIUS key is encrypted. For example:

```
FastIron(config)#radius-server key 1 abc
FastIron(config)#write terminal
...
radius-server host 1.2.3.5
radius key 1 $!2d
```

NOTE: Encryption of the RADIUS keys is done by default. The **0** parameter disables encryption. The **1** parameter is not required; it is provided for backwards compatibility.

Setting the Retransmission Limit

The **retransmit** parameter specifies the maximum number of retransmission attempts. When an authentication request times out, the Foundry software will retransmit the request up to the maximum number of retransmissions configured. The default retransmit value is 3 retries. The range of retransmit values is from 1 – 5.

To set the RADIUS retransmit limit:

```
FastIron(config)#radius-server retransmit 5
```

Syntax: radius-server retransmit <number>

Setting the Timeout Parameter

The **timeout** parameter specifies how many seconds the Foundry device waits for a response from the RADIUS server before either retrying the authentication request, or determining that the RADIUS server is unavailable and moving on to the next authentication method in the authentication-method list. The timeout can be from 1 – 15 seconds. The default is 3 seconds.

```
FastIron(config)#radius-server timeout 5
```

Syntax: radius-server timeout <number>

RADIUS over IPv6

Platform Support:

- FastIron X Series devices running software release 02.5.00 and later

Foundry devices support the ability to send RADIUS packets over an IPv6 network.

To enable the Foundry device to send RADIUS packets over IPv6, enter a command such as the following at the Global CONFIG level of the CLI:

```
FastIron(config)#radius-server host ipv6 3000::300
```

Syntax: radius-server host ipv6 <ipv6-host address>

The <ipv6-host address> is the IPv6 address of the RADIUS server. When you enter the IPv6 host address, you do not need to specify the prefix length. A prefix length of 128 is implied.

Configuring Authentication-Method Lists for RADIUS

You can use RADIUS to authenticate Telnet/SSH access and access to Privileged EXEC level and CONFIG levels of the CLI. When configuring RADIUS authentication, you create authentication-method lists specifically for these access methods, specifying RADIUS as the primary authentication method.

Within the authentication-method list, RADIUS is specified as the primary authentication method and up to six backup authentication methods are specified as alternates. If RADIUS authentication fails due to an error, the device tries the backup authentication methods in the order they appear in the list.

When you configure authentication-method lists for RADIUS, you must create a separate authentication-method list for Telnet or SSH CLI access and for CLI access to the Privileged EXEC level and CONFIG levels of the CLI.

To create an authentication-method list that specifies RADIUS as the primary authentication method for securing Telnet access to the CLI:

```
FastIron(config)#enable telnet authentication
FastIron(config)#aaa authentication login default radius local
```

The commands above cause RADIUS to be the primary authentication method for securing Telnet access to the CLI. If RADIUS authentication fails due to an error with the server, local authentication is used instead.

To create an authentication-method list that specifies RADIUS as the primary authentication method for securing access to Privileged EXEC level and CONFIG levels of the CLI:

```
FastIron(config)#aaa authentication enable default radius local none
```

The command above causes RADIUS to be the primary authentication method for securing access to Privileged EXEC level and CONFIG levels of the CLI. If RADIUS authentication fails due to an error with the server, local authentication is used instead. If local authentication fails, no authentication is used; the device automatically permits access.

Syntax: [no] aaa authentication enable | login default <method1> [<method2>] [<method3>] [<method4>] [<method5>] [<method6>] [<method7>]

The **web-server | enable | login** parameter specifies the type of access this authentication-method list controls. You can configure one authentication-method list for each type of access.

NOTE: If you configure authentication for Web management access, authentication is performed each time a page is requested from the server. When frames are enabled on the Web management interface, the browser sends an HTTP request for each frame. The Foundry device authenticates each HTTP request from the browser. To limit authentications to one per page, disable frames on the Web management interface.

The <method1> parameter specifies the primary authentication method. The remaining optional <method> parameters specify additional methods to try if an error occurs with the primary method. A method can be one of the values listed in the Method Parameter column in the following table.

Table 39.6: Authentication Method Values

Method Parameter	Description
line	Authenticate using the password you configured for Telnet access. The Telnet password is configured using the **enable telnet password…** command. See “Setting a Telnet Password” on page 39-13.
enable	Authenticate using the password you configured for the Super User privilege level. This password is configured using the **enable super-user-password…** command. See “Setting Passwords for Management Privilege Levels” on page 39-13.
local	Authenticate using a local user name and password you configured on the device. Local user names and passwords are configured using the **username…** command. See “Configuring a Local User Account” on page 39-21.
tacacs	Authenticate using the database on a TACACS server. You also must identify the server to the device using the **tacacs-server** command.
tacacs+	Authenticate using the database on a TACACS+ server. You also must identify the server to the device using the **tacacs-server** command.
radius	Authenticate using the database on a RADIUS server. You also must identify the server to the device using the **radius-server** command.
none	Do not use any authentication method. The device automatically permits access.

NOTE: For examples of how to define authentication-method lists for types of authentication other than RADIUS, see “Configuring Authentication-Method Lists” on page 39-57.

Entering Privileged EXEC Mode After a Telnet or SSH Login

By default, a user enters User EXEC mode after a successful login through Telnet or SSH. Optionally, you can configure the device so that a user enters Privileged EXEC mode after a Telnet or SSH login. To do this, use the following command:

```
FastIron(config)#aaa authentication login privilege-mode
```

Syntax: aaa authentication login privilege-mode

The user’s privilege level is based on the privilege level granted during login.

Configuring Enable Authentication to Prompt for Password Only

If Enable authentication is configured on the device, when a user attempts to gain Super User access to the Privileged EXEC and CONFIG levels of the CLI, by default he or she is prompted for a username and password. In this release, you can configure the Foundry device to prompt only for a password. The device uses the username entered at login, if one is available. If no username was entered at login, the device prompts for both username and password.

To configure the Foundry device to prompt only for a password when a user attempts to gain Super User access to the Privileged EXEC and CONFIG levels of the CLI:

```
FastIron(config)#aaa authentication enable implicit-user
```

Syntax: [no] aaa authentication enable implicit-user

Configuring RADIUS Authorization

Foundry devices support RADIUS authorization for controlling access to management functions in the CLI. Two kinds of RADIUS authorization are supported:

- Exec authorization determines a user's privilege level when they are authenticated
- Command authorization consults a RADIUS server to get authorization for commands entered by the user

Configuring Exec Authorization

When RADIUS exec authorization is performed, the Foundry device consults a RADIUS server to determine the privilege level of the authenticated user. To configure RADIUS exec authorization on the Foundry device, enter the following command:

```
FastIron(config)#aaa authorization exec default radius
```

Syntax: aaa authorization exec default radius | none

If you specify **none**, or omit the **aaa authorization exec** command from the device's configuration, no exec authorization is performed.

NOTE: If the **aaa authorization exec default radius** command exists in the configuration, following successful authentication the device assigns the user the privilege level specified by the foundry-privilege-level attribute received from the RADIUS server. If the **aaa authorization exec default radius** command does not exist in the configuration, then the value in the foundry-privilege-level attribute is ignored, and the user is granted Super User access.

Also note that in order for the **aaa authorization exec default radius** command to work, either the **aaa authentication enable default radius** command, or the **aaa authentication login privilege-mode** command must also exist in the configuration.

Configuring Command Authorization

When RADIUS command authorization is enabled, the Foundry device consults the list of commands supplied by the RADIUS server during authentication to determine whether a user can execute a command he or she has entered.

You enable RADIUS command authorization by specifying a privilege level whose commands require authorization. For example, to configure the Foundry device to perform authorization for the commands available at the Super User privilege level (that is; all commands on the device), enter the following command:

```
FastIron(config)#aaa authorization commands 0 default radius
```

Syntax: aaa authorization commands <privilege-level> default radius | tacacs+ | none

The <privilege-level> parameter can be one of the following:

- **0** – Authorization is performed (that is, the Foundry device looks at the command list) for commands available at the Super User level (all commands)
- **4** – Authorization is performed for commands available at the Port Configuration level (port-config and read-only commands)
- **5** – Authorization is performed for commands available at the Read Only level (read-only commands)

NOTE: RADIUS command authorization can be performed only for commands entered from Telnet or SSH sessions, or from the console. No authorization is performed for commands entered at the Web management interface or IronView Network Manager.

NOTE: Since RADIUS command authorization relies on the command list supplied by the RADIUS server during authentication, you cannot perform RADIUS authorization without RADIUS authentication.

Command Authorization and Accounting for Console Commands

The Foundry device supports command authorization and command accounting for CLI commands entered at the console. To configure the device to perform command authorization and command accounting for console commands, enter the following:

```
FastIron(config)#enable aaa console
```

Syntax: enable aaa console

CAUTION: If you have previously configured the device to perform command authorization using a RADIUS server, entering the **enable aaa console** command may prevent the execution of any subsequent commands entered on the console.

This happens because RADIUS command authorization requires a list of allowable commands from the RADIUS server. This list is obtained during RADIUS authentication. For console sessions, RADIUS authentication is performed only if you have configured Enable authentication and specified RADIUS as the authentication method (for example, with the **aaa authentication enable default radius** command). If RADIUS authentication is never performed, the list of allowable commands is never obtained from the RADIUS server. Consequently, there would be no allowable commands on the console.

Configuring RADIUS Accounting

Foundry devices support RADIUS accounting for recording information about user activity and system events. When you configure RADIUS accounting on a Foundry device, information is sent to a RADIUS accounting server when specified events occur, such as when a user logs into the device or the system is rebooted.

Configuring RADIUS Accounting for Telnet/SSH (Shell) Access

To send an Accounting Start packet to the RADIUS accounting server when an authenticated user establishes a Telnet or SSH session on the Foundry device, and an Accounting Stop packet when the user logs out:

```
FastIron(config)#aaa accounting exec default start-stop radius
```

Syntax: aaa accounting exec default start-stop radius | tacacs+ | none

Configuring RADIUS Accounting for CLI Commands

You can configure RADIUS accounting for CLI commands by specifying a privilege level whose commands require accounting. For example, to configure the Foundry device to perform RADIUS accounting for the commands available at the Super User privilege level (that is; all commands on the device), enter the following command:

```
FastIron(config)#aaa accounting commands 0 default start-stop radius
```

An Accounting Start packet is sent to the RADIUS accounting server when a user enters a command, and an Accounting Stop packet is sent when the service provided by the command is completed.

NOTE: If authorization is enabled, and the command requires authorization, then authorization is performed before accounting takes place. If authorization fails for the command, no accounting takes place.

Syntax: aaa accounting commands <privilege-level> default start-stop radius | tacacs | none

The <privilege-level> parameter can be one of the following:

- **0** – Records commands available at the Super User level (all commands)
- **4** – Records commands available at the Port Configuration level (port-config and read-only commands)

- **5** – Records commands available at the Read Only level (read-only commands)

Configuring RADIUS Accounting for System Events

You can configure RADIUS accounting to record when system events occur on the Foundry device. System events include rebooting and when changes to the active configuration are made.

The following command causes an Accounting Start packet to be sent to the RADIUS accounting server when a system event occurs, and a Accounting Stop packet to be sent when the system event is completed:

```
FastIron(config)#aaa accounting system default start-stop radius
```

Syntax: aaa accounting system default start-stop radius | tacacs+ | none

Configuring an Interface as the Source for All RADIUS Packets

You can designate the lowest-numbered IP address configured an Ethernet port, POS port, loopback interface, or virtual interface as the source IP address for all RADIUS packets from the Layer 3 Switch. Identifying a single source IP address for RADIUS packets provides the following benefits:

- If your RADIUS server is configured to accept packets only from specific links or IP addresses, you can use this feature to simplify configuration of the RADIUS server by configuring the Foundry device to always send the RADIUS packets from the same link or source address.
- If you specify a loopback interface as the single source for RADIUS packets, RADIUS servers can receive the packets regardless of the states of individual links. Thus, if a link to the RADIUS server becomes unavailable but the client or server can be reached through another link, the client or server still receives the packets, and the packets still have the source IP address of the loopback interface.

The software contains separate CLI commands for specifying the source interface for Telnet, TACACS/TACACS+, and RADIUS packets. You can configure a source interface for one or more of these types of packets.

To specify an Ethernet, loopback or virtual interface as the source for all RADIUS packets from the device, use the following CLI method. The software uses the lowest-numbered IP address configured on the port or interface as the source IP address for RADIUS packets originated by the device.

To specify the lowest-numbered IP address configured on a virtual interface as the device's source for all RADIUS packets, enter commands such as the following:

```
FastIron(config)#int ve 1
FastIron(config-vif-1)#ip address 10.0.0.3/24
FastIron(config-vif-1)#exit
FastIron(config)#ip radius source-interface ve 1
```

The commands in this example configure virtual interface 1, assign IP address 10.0.0.3/24 to the interface, then designate the interface as the source for all RADIUS packets from the Layer 3 Switch.

Syntax: ip radius source-interface ethernet [<slotnum>/]<portnum> | loopback <num> | ve <num>

The <slotnum> parameter is required on chassis devices.

The <portnum> parameter is a valid port number.

The <num> parameter is a loopback interface or virtual interface number. If you specify an Ethernet port, the <portnum> is the port's number (including the slot number, if you are configuring a device).

Displaying RADIUS Configuration Information

The **show aaa** command displays information about all TACACS/TACACS+ and RADIUS servers identified on the device. For example:

```
FastIron#show aaa
Tacacs+ key: foundry
Tacacs+ retries: 1
Tacacs+ timeout: 15 seconds
Tacacs+ dead-time: 3 minutes
Tacacs+ Server: 207.95.6.90 Port:49:
                opens=6 closes=3 timeouts=3 errors=0
                packets in=4 packets out=4
no connection

Radius key: networks
Radius retries: 3
Radius timeout: 3 seconds
Radius dead-time: 3 minutes
Radius Server:  207.95.6.90 Auth Port=1645 Acct Port=1646:
                opens=2 closes=1 timeouts=1 errors=0
                packets in=1 packets out=4
no connection
```

The following table describes the RADIUS information displayed by the **show aaa** command.

Table 39.7: Output of the show aaa command for RADIUS

Field	Description
Radius key	The setting configured with the **radius-server key** command. At the Super User privilege level, the actual text of the key is displayed. At the other privilege levels, a string of periods (....) is displayed instead of the text.
Radius retries	The setting configured with the **radius-server retransmit** command.
Radius timeout	The setting configured with the **radius-server timeout** command.
Radius dead-time	The setting configured with the **radius-server dead-time** command.
Radius Server	For each RADIUS server, the IP address, and the following statistics are displayed: Auth Port RADIUS authentication port number (default 1645) Acct Port RADIUS accounting port number (default 1646) opens Number of times the port was opened for communication with the server closes Number of times the port was closed normally timeouts Number of times port was closed due to a timeout errors Number of times an error occurred while opening the port packets in Number of packets received from the server packets out Number of packets sent to the server
connection	The current connection status. This can be “no connection” or “connection active”.

The **show web** command displays the privilege level of Web management interface users. For example:

```
FastIron#show web
User                                Privilege       IP address
set                                         0       192.168.1.234
```

Syntax: show web

Configuring Authentication-Method Lists

To implement one or more authentication methods for securing access to the device, you configure authentication-method lists that set the order in which the authentication methods are consulted.

In an authentication-method list, you specify the access method (Telnet, Web, SNMP, and so on) and the order in which the device tries one or more of the following authentication methods:

- Local Telnet login password
- Local password for the Super User privilege level
- Local user accounts configured on the device
- Database on a TACACS or TACACS+ server
- Database on a RADIUS server
- No authentication

NOTE: The TACACS/TACACS+, RADIUS, and Telnet login password authentication methods are not supported for SNMP access.

NOTE: To authenticate Telnet access to the CLI, you also must enable the authentication by entering the **enable telnet authentication** command at the global CONFIG level of the CLI. You cannot enable Telnet authentication using the Web management interface.

NOTE: You do not need an authentication-method list to secure access based on ACLs or a list of IP addresses. See “Using ACLs to Restrict Remote Access” on page 39-4 or “Restricting Remote Access to the Device to Specific IP Addresses” on page 39-6.

In an authentication-method list for a particular access method, you can specify up to seven authentication methods. If the first authentication method is successful, the software grants access and stops the authentication process. If the access is rejected by the first authentication method, the software denies access and stops checking.

However, if an error occurs with an authentication method, the software tries the next method on the list, and so on. For example, if the first authentication method is the RADIUS server, but the link to the server is down, the software will try the next authentication method in the list.

NOTE: If an authentication method is working properly and the password (and user name, if applicable) is not known to that method, this is not an error. The authentication attempt stops, and the user is denied access.

The software will continue this process until either the authentication method is passed or the software reaches the end of the method list. If the Super User level password is not rejected after all the access methods in the list have been tried, access is granted.

Configuration Considerations for Authentication-Method Lists

- For CLI access, you must configure authentication-method lists if you want the device to authenticate access using local user accounts or a RADIUS server. Otherwise, the device will authenticate using only the locally based password for the Super User privilege level.
- When no authentication-method list is configured specifically for Web management access, the device performs authentication using the SNMP community strings:
    - For read-only access, you can use the user name “get” and the password “public”. The default read-only community string is “public”.
    - There is no default read-write community string. Thus, by default, you cannot open a read-write management session using the Web management interface. You first must configure a read-write community string using the CLI. Then you can log on using “set” as the user name and the read-write community string you configure as the password. See “Configuring TACACS/TACACS+ Security” on page 39-27.
- If you configure an authentication-method list for Web management access and specify “local” as the primary authentication method, users who attempt to access the device using the Web management interface must supply a user name and password configured in one of the local user accounts on the device. The user ***cannot*** access the device by entering “set” or “get” and the corresponding SNMP community string.
- For devices that can be managed using IronView Network Manager, the default authentication method (if no authentication-method list is configured for SNMP) is the CLI Super User level password. If no Super User level password is configured, then access through IronView Network Manager is not authenticated. To use local user accounts to authenticate access through IronView Network Manager, configure an authentication-method list for SNMP access and specify “local” as the primary authentication method.

Examples of Authentication-Method Lists

Example 1

The following example shows how to configure authentication-method lists for the Web management interface, IronView Network Manager, and the Privileged EXEC and CONFIG levels of the CLI. In this example, the primary authentication method for each is “local”. The device will authenticate access attempts using the locally configured user names and passwords first.

To configure an authentication-method list for the Web management interface, enter a command such as the following:

```
FastIron(config)#aaa authentication web-server default local
```

This command configures the device to use the local user accounts to authenticate access to the device through the Web management interface. If the device does not have a user account that matches the user name and password entered by the user, the user is not granted access.

To configure an authentication-method list for IronView Network Manager, enter a command such as the following:

```
FastIron(config)#aaa authentication snmp-server default local
```

This command configures the device to use the local user accounts to authenticate access attempts through any network management software, such as IronView Network Manager.

To configure an authentication-method list for the Privileged EXEC and CONFIG levels of the CLI, enter the following command:

```
FastIron(config)#aaa authentication enable default local
```

This command configures the device to use the local user accounts to authenticate attempts to access the Privileged EXEC and CONFIG levels of the CLI.

Example 2

To configure the device to consult a RADIUS server first to authenticate attempts to access the Privileged EXEC and CONFIG levels of the CLI, then consult the local user accounts if the RADIUS server is unavailable, enter the following command:

```
FastIron(config)#aaa authentication enable default radius local
```

Syntax: [no] aaa authentication snmp-server | web-server | enable | login default <method1> [<method2>] [<method3>] [<method4>] [<method5>] [<method6>] [<method7>]

The **snmp-server | web-server | enable | login** parameter specifies the type of access this authentication-method list controls. You can configure one authentication-method list for each type of access.

NOTE: TACACS/TACACS+ and RADIUS are supported only with the **enable** and **login** parameters.

The <method1> parameter specifies the primary authentication method. The remaining optional <method> parameters specify additional methods to try if an error occurs with the primary method. A method can be one of the values listed in the Method Parameter column in the following table.

Table 39.8: Authentication Method Values

Method Parameter	Description
line	Authenticate using the password you configured for Telnet access. The Telnet password is configured using the **enable telnet password…** command. See “Setting a Telnet Password” on page 39-13.
enable	Authenticate using the password you configured for the Super User privilege level. This password is configured using the **enable super-user-password…** command. See “Setting Passwords for Management Privilege Levels” on page 39-13.
local	Authenticate using a local user name and password you configured on the device. Local user names and passwords are configured using the **username…** command. See “Configuring a Local User Account” on page 39-21.
tacacs	Authenticate using the database on a TACACS server. You also must identify the server to the device using the **tacacs-server** command.
tacacs+	Authenticate using the database on a TACACS+ server. You also must identify the server to the device using the **tacacs-server** command.
radius	Authenticate using the database on a RADIUS server. You also must identify the server to the device using the **radius-server** command. See “Configuring RADIUS Security” on page 39-42.
none	Do not use any authentication method. The device automatically permits access.

TCP Flags - Edge Port Security

This release supports edge port security on FastIron GS and FastIron LS devices. This feature works in combination with IP ACL rules, and supports all 6 TCP flags present in the offset 13 of the TCP header:

- +|- urg = Urgent
- +|- ack = Acknowledge
- +|- psh = Push
- +|- rst = Reset
- +|- syn = Synchronize
- +|- fin = Finish

TCP flags can be combined with other ACL functions (such as dscp-marking and traffic policies), giving you greater flexibility when designing ACLs.

The TCP flags feature offers two options, match-all and match-any:

- **Match-any** - Indicates that incoming TCP traffic must be matched against any of the TCP flags configured as part of the match-any ACL rule. In CAM hardware, the number of ACL rules will match the number of configured flags.
- **Match-all** - Indicates that incoming TCP traffic must be matched against all of the TCP flags configured as part of the match-all ACL rule. In CAM hardware, there will be only one ACL rule for all configured flags. For example:

```
FastIron(config-ext-nacl)#permit tcp 1.1.1.1 0.0.0.255 eq 100 2.2.2.2 0.0.0.255 eq
300 match-all +urg +ack +syn -rst
```

This command configures a single rule in CAM hardware. This rule will contain all of the configured TCP flags (urg, ack, syn, and rst).

Using TCP Flags in Combination with Other ACL Features

The TCP Flags feature has the added capability of being combined with other ACL features. For example:

```
FastIron(config-ext-nacl)#permit tcp any any match-all +urg +ack +syn -rst traffic-
policy test
```

This command configures the ACL to match incoming traffic with the TCP Flags urg, ack, and syn and also to apply the traffic policy (rate, limit, etc.) to the matched traffic.

```
FastIron(config-ext-nacl)#permit tcp any any match-all +urg +ack +syn -rst tos
normal
```

This command configures the ACL to match incoming traffic with the flags urg, ack, and syn, and also sets the tos bit to normal when the traffic exits the device.

NOTE: TCP Flags combines the functionality of older features such as TCP Syn Attack and TCP Establish. Avoid configuring these older features on a port where you have configured TCP Flags. TCP Flags can perform all of the functions of TCP Syn Attack and TCP Establish, and more. However, if TCP Syn Attack is configured on a port along with TCP Flags, TCP Syn Attack will take precedence.

NOTE: If an ACL clause with match-any exists, and the system runs out of CAM, if the total number of TCP rules to TCP Flags will not fit within 1021 entries (the maximum rules allowed per device), then none of the TCP Flag rules will be programmed into the CAM hardware.

NOTE: If a range option and match-any TCP-flags are combined in the same ACL, the total number of rules will be calculated as: Total number of rules in CAM hardware = (number of rules for range)* (number of rules for match-any TCP-flags).

Chapter 40

Configuring SSHv1 and SCP

Secure Shell (SSH) version 1 is supported in the following releases:

- Pre-release 03.0.00 software for the FastIron X Series devices
- Pre-release 02.4.00 for the FastIron GS

In succeeding releases, SSHv1 is no longer supported and SSH version 2 (SSHv2) is supported. For SSHv2 configuration procedures, see the chapter "Configuring SSHv2 and SCP" on page 41-1.

Secure Shell (SSH) is a mechanism for allowing secure remote access to management functions on a Foundry device. SSH provides a function similar to Telnet. Users can log into and configure the device using a publicly or commercially available SSH client program, just as they can with Telnet. However, unlike Telnet which provides no security, SSH provides a secure, encrypted connection to the device.

SSH supports Arcfour, IDEA, Blowfish, DES (56-bit) and Triple DES (168-bit) data encryption methods. Nine levels of data compression are available. You can configure your SSH client to use any one of these data compression levels when connecting to a Foundry device.

Foundry devices also support Secure Copy (SCP) for securely transferring files between a Foundry device and SCP-enabled remote hosts. See "Using Secure Copy with SSHv1" on page 40-10 for more information.

Configuring SSHv1

Foundry's implementation of SSH supports two kinds of user authentication:

- ***RSA challenge-response authentication***, where a collection of public keys are stored on the device. Only clients with a private key that corresponds to one of the stored public keys can gain access to the device using SSH.
- ***Password authentication***, where users attempting to gain access to the device using an SSH client are authenticated with passwords stored on the device or on a TACACS/TACACS+ or RADIUS server

Both kinds of user authentication are enabled by default. You can configure the device to use one or both of them.

To configure Secure Shell on a Foundry device, do the following:

1. Set the Foundry device's host name and domain name.
2. Generate a host RSA public and private key pair for the device.
3. Configure RSA challenge-response authentication.
4. Set optional parameters.

You can also view information about active SSH connections on the device as well as terminate them.

Setting the Host Name and Domain Name

If you have not already done so, establish a host name and domain name for the Foundry device. For example:

```
FastIron(config)#hostname Fesx424Router
FastIron(config)#ip dns domain-name foundrynet.com
```

Syntax: hostname <name>

Syntax: ip dns domain-name <name>

Generating a Host I Key Pair

When SSH is configured, a public and private ***host RSA key pair*** is generated for the Foundry device. The SSH server on the Foundry device uses this host RSA key pair, along with a dynamically generated ***server RSA key pair***, to negotiate a session key and encryption method with the client trying to connect to it.

The host RSA key pair is stored in the Foundry device's system-config file. Only the public key is readable. The public key should be added to a "known hosts" file (for example, $HOME/.ssh/known_hosts on UNIX systems) on the clients who want to access the device. Some SSH client programs add the public key to the known hosts file automatically; in other cases, you must manually create a known hosts file and place the Foundry device's public key in it. See "Providing the Public Key to Clients" on page 40-3 for an example of what to place in the known hosts file.

To generate a public and private RSA host key pair for the most Foundry devices, enter the following commands:

```
FastIron(config)#crypto key generate rsa
FastIron(config)#write memory
```

The **crypto key generate [rsa]** command places an RSA host key pair in the running-config file and enables SSH on the device.

To disable SSH, you must delete the RSA host key pair. To do this in SSHv1, enter the following commands:

```
FastIron(config)#crypto key zeroize rsa
FastIron(config)#write memory
```

The **crypto key zeroize [rsa]** command deletes the RSA host key pair in the running-config file and disables SSH on the device.

Syntax: crypto key generate | zeroize rsa

You can optionally configure the Foundry device to hide the RSA host key pair in the running-config file. To do this, enter the following command:

```
FastIron#ssh no-show-host-keys
```

Syntax: ssh no-show-host-keys

After entering the **ssh no-show-host-keys** command, you can display the RSA host key pair in the running-config file with the following command:

```
FastIron#ssh show-host-keys
```

Syntax: ssh show-host-keys

Configuration Notes

- If the Foundry device does not have internal memory, the RSA host key pair is stored in the startup-config file. In this case, the **ssh show-host-keys** command allows the RSA host key pair to be viewed, and the **ssh no-show-host-keys** command prevents the RSA host key pair from being viewed.
- If an RSA host key pair is stored in internal memory on the Foundry device, it is used even if the startup-config file contains a different RSA host key pair.
- If no RSA host key pair is stored in internal memory, but the startup-config file contains an RSA host key pair, the key pair in the startup-config file is used. If you later generate an RSA host key pair with the **crypto key generate** command, the new key pair takes effect only after you store it in internal memory with the **write memory** command and reboot the Foundry device.

- If no RSA host key pair is stored in internal memory, and the startup-config file contains an RSA host key pair, the first time you enter the **write memory** command, it will save the RSA host key pair in the startup-config file to internal memory and remove it from the startup-config file.
- If no RSA host key pair is stored in internal memory, the startup-config file contains an RSA host key pair, and you generate an RSA host key pair with the **crypto key generate** command, the new pair is stored in internal memory the first time you enter the **write memory** command.
- The **crypto key zeroize** command disables the currently active RSA host key pair. If you subsequently enter the **write memory** command without generating another RSA host key pair, the RSA host key pair stored in internal memory is removed.
- If you enter the **ssh no-show-host-keys** command to hide the RSA host key pair in the running-config file, then reload the software, the RSA host key pair is once again visible in the running-config file. The setting to hide the RSA host key pair is not carried across software reloads.
- In a configuration using redundant management modules, if the active module has an RSA host key pair, but the standby module does not, the RSA host key pair is not carried over when switchover occurs. You must create an RSA host key pair on the standby module manually.
- The SSH key generation process causes UDLD-enabled interfaces to go down instantaneously. This in turn requires the re-convergence of the route tables on the routers across the network. Non-UDLD-enabled interfaces do not experience this issue.

Providing the Public Key to Clients

If you are using SSH to connect to a Foundry device from a UNIX system, you may need to add the Foundry device's public key to a "known hosts" file; for example, $HOME/.ssh/known_hosts. The following is an example of an entry in a known hosts file:

```
10.10.20.10 1024 37 118771881862677030464851288737258046856031640635887679230111
84247022636175804896633384620574930068397650231698985431857279323745963240790218
03229084221453472515782437007702806627934784079949643404159653290224014833380339
09542147367974638560060162945329307563502804231039654388220432832662804242569361
58342816331
```

In this example, 10.10.20.10 is the IP address of an SSH-enabled Foundry Layer 2 Switch or Layer 3 Switch. The second number, 1024, is the size of the host key, and the third number, 37, is the encoded public exponent. The remaining text is the encoded modulus.

Configuring RSA Challenge-Response Authentication

With RSA challenge-response authentication, a collection of clients' public keys are stored on the Foundry device. Clients are authenticated using these stored public keys. Only clients that have a private key that corresponds to one of the stored public keys can gain access to the device using SSH.

When RSA challenge-response authentication is enabled, the following events occur when a client attempts to gain access to the device using SSH:

1. The client sends its public key to the Foundry device.
2. The Foundry device compares the client's public key to those stored in memory.
3. If there is a match, the Foundry device uses the public key to encrypt a random sequence of bytes.
4. The Foundry device sends these encrypted bytes to the client.
5. The client uses its private key to decrypt the bytes.
6. The client sends the decrypted bytes back to the Foundry device.
7. The Foundry device compares the decrypted bytes to the original bytes it sent to the client. If the two sets of bytes match, it means that the client's private key corresponds to an authorized public key, and the client is authenticated.

Setting up RSA challenge-response authentication consists of the following steps:

1. Importing authorized public keys into the Foundry device.
2. Enabling RSA challenge response authentication

Importing Authorized Public Keys into the Foundry Device

SSH clients that support RSA authentication normally provide a utility to generate an RSA key pair. The private key is usually stored in a password-protected file on the local host; the public key is stored in another file and is not protected. You should collect one public key from each client to be granted access to the Foundry device and place all of these keys into one file. This public key file is imported into the Foundry device.

The following is an example of a public key file containing two public keys:

```
1024 65537 162566050678380006149460550286514061230306797782065166110686648548574
94957339232259963157379681924847634614532742178652767231995746941441604714682680
00644536790333304202912490569077182886541839656556769025432881477252978135927821
67540629478392662275128774861815448523997023618173312328476660721888873946758201
 user@csp_client
1024 35 152676199889856769693556155614587291553826312328095300428421494164360924
76207475545234679268443233762295312979418833525975695775705101805212541008074877
26586119857422702897004112168852145074087969840642408451742714558592361693705908
74837875599405503479603024287131312793895007927438074972787423695977635251943 ro
ot@unix_machine
```

You can import the authorized public keys into the active configuration by loading them from a file on a TFTP server. Once the authorized public keys are loaded, you can optionally save them to the startup-config file. If you import a public key file from a TFTP server or PCMCIA flash card, the file is automatically loaded into the active configuration the next time the device is booted.

Foundry devices support Secure Copy (SCP) for securely transferring files between hosts on a network. Note that when you copy files using SCP, you enter the commands on the SCP-enabled client, rather than the console on the Foundry device.

For example, to copy a public key file called pkeys.txt from an SCP-enabled client, enter a command such as the following on the SCP-enabled client:

```
C:\> scp c:\pkeys.txt user@FastIron:a:/pkeys.txt
```

If password authentication is enabled for SSH, the user will be prompted for a password in order to copy the file. See “Using Secure Copy with SSHv1” on page 40-10 for more information on SCP.

After the file is loaded onto the TFTP server or PCMCIA flash card, it can be imported into the active configuration each time the device is booted.

To cause a public key file called pkeys.txt to be loaded from a TFTP server each time the Foundry device is booted, enter a command such as the following:

```
FastIron(config)#ip ssh pub-key-file tftp 192.168.1.234 pkeys.txt
```

Syntax: ip ssh pub-key-file tftp <tftp-server-ip-addr> <filename>

The <tftp-server-ip-addr> variable is the IP address of the tftp server that contains the public key file that you want to import into the Foundry device.

The <filename> variable is the name of the dsa public key file that you want to import into the Foundry device.

To display the currently loaded public keys, enter the following command:

```
FastIron#show ip client-pub-key

1024 65537 162566050678380006149460550286514061230306797782065166110686648548574
949573392322599631573796819248476346145327421786527672319957469414416047146826800
0644536790333304202912490569077182886541839656556769025432881477252978135927821
67540629478392662275128774861815448523997023618173312328476660721888873946758201
 user@csp_client

1024 35 152676199889856769693556155614587291553826312328095300428421494164360924
76207475545234679268443233762295312979418833525975695775705101805212541008074877
26586119857422702897004112168852145074087969840642408451742714558592361693705908
74837875599405503479603024287131312793895007927438074972787423695977635251943 ro
ot@unix_machine

There are 2 authorized client public keys configured
```

Syntax: show ip client-pub-key

To clear the public keys from the active configuration, enter the following command:

```
FastIron#clear public-key
```

Syntax: clear public-key

To reload the public keys from the file on the TFTP server or PCMCIA flash card, enter the following command:

```
FastIron(config)#ip ssh pub-key-file reload
```

Syntax: ip ssh pub-key-file reload

Once the public keys are part of the active configuration, you can make them part of the startup-config file. The startup-config file can contain a maximum of 10 public keys. If you want to store more than 10 public keys, keep them in a file on a TFTP server or PCMCIA flash card, where they will be loaded into the active configuration when the device is booted.

To make the public keys in the active configuration part of the startup-config file, enter the following commands:

```
FastIron(config)#ip ssh pub-key-file flash-memory
FastIron(config)#write memory
```

Syntax: ip ssh pub-key-file flash-memory

To clear the public keys from the startup-config file (if they are located there), enter the following commands:

```
FastIron#clear public-key
FastIron#write memory
```

Enabling RSA Challenge-Response Authentication

RSA challenge-response authentication is enabled by default. You can disable or re-enable it manually.

To enable RSA challenge-response authentication:

```
FastIron(config)#ip ssh rsa-authentication yes
```

To disable RSA challenge-response authentication:

```
FastIron(config)#ip ssh rsa-authentication no
```

Syntax: ip ssh rsa-authentication yes | no

Setting Optional Parameters

You can adjust the following SSH settings on the Foundry device:

- The number of SSH authentication retries
- The server RSA key size
- The user authentication method the Foundry device uses for SSH connections
- Whether the Foundry device allows users to log in without supplying a password
- The port number for SSH connections
- The SSH login timeout value
- A specific interface to be used as the source for all SSH traffic from the device
- The maximum idle time for SSH sessions

Setting the Number of SSH Authentication Retries

By default, the Foundry device attempts to negotiate a connection with the connecting host three times. The number of authentication retries can be changed to between 1 – 5.

For example, the following command changes the number of authentication retries to 5:

```
FastIron(config)#ip ssh authentication-retries 5
```

Syntax: ip ssh authentication-retries <number>

Setting the Server RSA Key Size

The default size of the dynamically generated server RSA key is 768 bits. The size of the server RSA key can be between 512 – 896 bits.

For example, the following command changes the server RSA key size to 896 bits:

```
FastIron(config)#ip ssh key-size 896
```

Syntax: ip ssh key-size <number>

NOTE: The **ip ssh key-size** command is not applicable to SSHv2 implementation.

NOTE: The size of the ***host*** RSA key that resides in the system-config file is always 1024 bits and cannot be changed.

Deactivating User Authentication

After the SSH server on the Foundry device negotiates a session key and encryption method with the connecting client, user authentication takes place. Foundry's implementation of SSH supports RSA challenge-response authentication and password authentication.

With RSA challenge-response authentication, a collection of clients' public keys are stored on the Foundry device. Clients are authenticated using these stored public keys. Only clients that have a private key that corresponds to one of the stored public keys can gain access to the device using SSH.

With password authentication, users are prompted for a password when they attempt to log into the device (provided empty password logins are not allowed; see "Enabling Empty Password Logins" on page 40-7). If there is no user account that matches the user name and password supplied by the user, the user is not granted access.

You can deactivate one or both user authentication methods for SSH. Note that deactivating both authentication methods essentially disables the SSH server entirely.

To disable RSA challenge-response authentication:

```
FastIron(config)#ip ssh rsa-authentication no
```

Syntax: ip ssh rsa-authentication no | yes

To deactivate password authentication:

```
FastIron(config)#ip ssh password-authentication no
```

Syntax: ip ssh password-authentication no | yes

Enabling Empty Password Logins

By default, empty password logins are not allowed. This means that users with an SSH client are always prompted for a password when they log into the device. To gain access to the device, each user must have a user name and password. Without a user name and password, a user is not granted access. See “Setting up Local User Accounts” on page 39-17 for information on setting up user names and passwords on Foundry devices.

If you enable empty password logins, users are ***not*** prompted for a password when they log in. Any user with an SSH client can log in without being prompted for a password.

To enable empty password logins:

```
FastIron(config)#ip ssh permit-empty-passwd yes
```

Syntax: ip ssh permit-empty-passwd no | yes

Setting the SSH Port Number

By default, SSH traffic occurs on TCP port 22. You can change this port number. For example, the following command changes the SSH port number to 2200:

```
FastIron(config)#ip ssh port 2200
```

Note that if you change the default SSH port number, you must configure SSH clients to connect to the new port. Also, you should be careful not to assign SSH to a port that is used by another service. If you change the SSH port number, Foundry recommends that you change it to a port number greater than 1024.

Syntax: ip ssh port <number>

Setting the SSH Login Timeout Value

When the SSH server attempts to negotiate a session key and encryption method with a connecting client, it waits a maximum of 120 seconds for a response from the client. If there is no response from the client after 120 seconds, the SSH server disconnects. You can change this timeout value to between 1 – 120 seconds. For example, to change the timeout value to 60 seconds:

```
FastIron(config)#ip ssh timeout 60
```

Syntax: ip ssh timeout <seconds>

Designating an Interface as the Source for All SSH Packets

You can designate a loopback interface, virtual interface, or Ethernet port as the source for all SSH packets from the device. The software uses the IP address with the numerically lowest value configured on the port or interface as the source IP address for SSH packets originated by the device.

NOTE: When you specify a single SSH source, you can use only that source address to establish SSH management sessions with the Foundry device.

To specify the numerically lowest IP address configured on a loopback interface as the device’s source for all SSH packets, enter commands such as a the following:

```
FastIron(config)#int loopback 2
FastIron(config-lbif-2)#ip address 10.0.0.2/24
FastIron(config-lbif-2)#exit
FastIron(config)#ip ssh source-interface loopback 2
```

The commands in this example configure loopback interface 2, assign IP address 10.0.0.2/24 to the interface, then designate the interface as the source for all SSH packets from the Layer 3 Switch.

Syntax: ip ssh source-interface ethernet [<slotnum>/]<portnum> | loopback <num> | ve <num>

The <slotnum> parameter is required on chassis devices.

The <portnum> parameter is a valid port number.

The <num> parameter is a loopback interface or virtual interface number. If you specify an Ethernet port, the <portnum> is the port's number (including the slot number, if you are configuring a Chassis device). For example:

```
FastIron(config)#interface ethernet 1/4
FastIron(config-if-1/4)#ip address 209.157.22.110/24
FastIron(config-if-1/4)#exit
FastIron(config)#ip ssh source-interface ethernet 1/4
```

Configuring Maximum Idle Time for SSH Sessions

By default, SSH sessions do not time out. Optionally, you can set the amount of time an SSH session can be inactive before the Foundry device closes it. For example, to set the maximum idle time for SSH sessions to 30 minutes:

```
FastIron(config)#ip ssh idle-time 30
```

Syntax: ip ssh idle-time <minutes>

If an established SSH session has no activity for the specified number of minutes, the Foundry device closes it. An idle time of 0 minutes (the default value) means that SSH sessions never timeout. The maximum idle time for SSH sessions is 240 minutes.

Displaying SSH Connection Information

Up to five SSH connections can be active on the Foundry device. To display information about SSH connections, enter the following command:

```
FastIron#show ip ssh
Connection     Version     Encryption     State     Username
    1            1.5         ARCFOUR        0x82      neville
    2            1.5          IDEA          0x82      lynval
    3            1.5          3DES          0x82      terry
    4            1.5          AES         0x00
    5              1.5          none          0x00
```

Syntax: show ip ssh

This display shows the following information about the active SSH connections:

Table 40.1: SSH Connection Information

This Field...	Displays...
Connection	The SSH connection ID. This can be from 1 – 5.
Version	The SSH version number. This should always be 1.5.
Encryption	The encryption method used for the connection. This can be IDEA, ARCFOUR, DES, 3DES, AES, or BLOWFISH.

Table 40.1: SSH Connection Information (Continued)

This Field...	Displays...
State	The connection state. This can be one of the following: 0x00 Server started to send version number to client. 0x01 Server sent version number to client. 0x02 Server received version number from client. 0x20 Server sent public key to client. 0x21 Server is waiting for client’s session key. 0x22 Server received session key from client. 0x23 Server is verifying client’s session key. 0x24 Client’s session key is verified. 0x25 Server received client’s name. 0x40 Server is authenticating client. 0x41 Server is continuing to authenticate client after one or more failed attempts. 0x80 Server main loop started after successful authentication. 0x81 Server main loop sent a message to client. 0x82 Server main loop received a message from client.
Username	The user name for the connection.

The **show who** command also displays information about SSH connections. For example:

```
FastIron#show who
Console connections:
 established, active
Telnet connections:
 1 closed
 2 closed
 3 closed
 4 closed
 5 closed
SSH connections:
 1 established, client ip address 209.157.22.8
 16 seconds in idle
 2 established, client ip address 209.157.22.21
 42 seconds in idle
 3 established, client ip address 209.157.22.68
 49 seconds in idle
 4 closed
 5 closed
```

Syntax: show who

To terminate one of the active SSH connections, enter the following command:

```
FastIron#kill ssh 1
```

Syntax: kill ssh <connection-id>

Sample SSH Configuration

The following is a sample SSH configuration for a Foundry device.

```
hostname FastIron
ip dns domain-name foundrynet.com
!
aaa authentication login default local
username neville password .....
username lynval password .....
username terry password .....
!
ip ssh permit-empty-passwd no
!
ip ssh pub-key-file tftp 192.168.1.234 pkeys.txt
!
crypto key generate rsa public_key "1024 35 144460146631716543532035011163035196
41193195125205894452637462409522275505020845087302985209960346239172995676329357
24777530188666267898195648253181551624681394520681672610828188310413962242301296
26883937176769776184984093100984017075369387071006637966650877224677979486802651
458324218055083313313948534902409 FastIron@foundrynet.com"
!
crypto key generate rsa private_key "*************************"
!
ip ssh authentication-retries 5
```

This **aaa authentication login default local** command configures the device to use the local user accounts to authenticate users attempting to log in.

Three user accounts are configured on the device. The **ip ssh permit-empty-passwd no** command causes users always to be prompted for a password when they attempt to establish an SSH connection. Since the device uses local user accounts for authentication, only these three users are allowed to connect to the device using SSH.

The **ip ssh pub-key-file tftp** command causes a public key file called pkeys.txt to be loaded from a TFTP server at 192.168.1.234. To gain access to the Foundry device using SSH, a user must have a private key that corresponds to one of the public keys in this file.

The **crypto key generate rsa public_key** and **crypto key generate rsa private_key** statements are both generated by the **crypto key generate rsa** command. By default, the RSA host key pair appears in the running-config file, but not in the startup-config file. You can optionally configure the Foundry device to hide the RSA host key pair in the running-config file with the **ssh no-show-host-keys** command. The actual private key is never visible in either the running-config file or the startup-config file.

You may need to copy the public key to a “known hosts” file (for example, $HOME/.ssh/known_hosts on UNIX systems) on the clients who want to access the device. See “Providing the Public Key to Clients” on page 40-3 for an example of what to place in the known hosts file.

The **ip ssh authentication-retries 5** command sets the number of times the Foundry device attempts to negotiate a connection with the connecting host to 5.

Using Secure Copy with SSHv1

Secure Copy (SCP) uses security built into SSH to transfer image and configuration files to and from the device. SCP automatically uses the authentication methods, encryption algorithm, and data compression level configured for SSH. For example, if password authentication is enabled for SSH, the user is prompted for a user name and password before SCP allows a file to be transferred. No additional configuration is required for SCP on top of SSH.

You can use SCP to copy files on the Foundry device, including the startup configuration and running configuration files, to or from an SCP-enabled remote host.

SCP is enabled by default and can be disabled. To disable SCP, enter the following command:

```
FastIron(config)# ip ssh scp disable
```

Syntax: ip ssh scp disable | enable

NOTE: If you disable SSH, SCP is also disabled.

The following are examples of using SCP to transfer files from and to a Foundry device.

NOTE: When using SCP, you enter the **scp** commands on the SCP-enabled client, rather than the console on the Foundry device.

NOTE: Certain SCP client options, including -p and -r, are ignored by the SCP server on the Foundry device. If an option is ignored, the client is notified.

To copy a configuration file (c:\cfg\foundry.cfg) to the running configuration file on a Foundry device at 192.168.1.50 and log in as user terry, enter the following command on the SCP-enabled client:

```
C:\> scp c:\cfg\foundry.cfg terry@192.168.1.50:runConfig
```

If password authentication is enabled for SSH, the user is prompted for user terry’s password before the file transfer takes place.

To copy the configuration file to the startup configuration file:

```
C:\> scp c:\cfg\foundry.cfg terry@192.168.1.50:startConfig
```

To copy the running configuration file on a Foundry device to a file called c:\cfg\fdryrun.cfg on the SCP-enabled client:

```
C:\> scp terry@192.168.1.50:runConfig c:\cfg\fdryrun.cfg
```

To copy the startup configuration file on a Foundry device to a file called c:\cfg\fdrystart.cfg on the SCP-enabled client:

```
C:\> scp terry@192.168.1.50:startConfig c:\cfg\fdrystart.cfg
```

Chapter 41

Configuring SSHv2 and SCP

Secure Shell (SSH) is a mechanism for allowing secure remote access to management functions on a Foundry device. SSH provides a function similar to Telnet. Users can log into and configure the device using a publicly or commercially available SSH client program, just as they can with Telnet. However, unlike Telnet, which provides no security, SSH provides a secure, encrypted connection to the device.

Foundry's SSHv2 implementation is compatible with all versions of the SSHv2 protocol (2.1, 2.2, and so on). At the beginning of an SSH session, the Foundry device negotiates the version of SSHv2 to be used. The highest version of SSHv2 supported by both the Foundry device and the client is the version that is used for the session. Once the SSHv2 version is negotiated, the encryption algorithm with the highest security ranking is selected to be used for the session.

Foundry devices also support Secure Copy (SCP) for securely transferring files between a Foundry device and SCP-enabled remote hosts.

NOTE: With the introduction of SSHv2, SSHv1 is no longer supported. For example, if you are running a FastIron X Series release prior to 03.0.00 and are using SSHv1, your keys will no longer be valid when you upgrade to FastIron X Series 03.0.00 or later. You must obtain new keys for SSHv2.

NOTE: The SSH feature includes software that is copyright Allegro Software Development Corporation.

SSH Version 2 Support

Platform Support:

- FGS devices running software release 02.4.00 and later
- FLS devices running software release 03.0.00 and later
- FESX/FSX/FWSX devices running software release 03.0.00 and later

Beginning with these releases, SSHv2 is supported in the Layer 2 and Layer 3 codes, and SSH version 1 (SSHv1) is no longer supported. Releases prior to those listed above support SSHv1. See "Configuring SSHv1 and SCP" on page 40-1.

SSHv2 is a substantial revision of Secure Shell, comprising the following hybrid protocols and definitions:

- SSH Transport Layer Protocol
- SSH Authentication Protocol

- SSH Connection Protocol
- SECSH Public Key File Format
- SSH Fingerprint Format
- SSH Protocol Assigned Numbers
- SSH Transport Layer Encryption Modes
- SCP/SFTP/SSH URI Format

Tested SSHv2 Clients

The following SSH clients have been tested with SSHv2:

- SSH Secure Shell 3.2.3
- Van Dyke SecureCRT 4.0 and 4.1
- F-Secure SSH Client 5.3 and 6.0
- PuTTY 0.54 and 0.56
- OpenSSH 3.5_p1 and 3.6.1p2
- Solaris Sun-SSH-1.0

NOTE: The FastIron GS and FastIron LS devices support client public key sizes of 1024 bytes or less. This also applies to FastIron X Series devices running software release 03.0.00 or later.

Supported Encryption Algorithms for SSHv2

Platform Support: FGS/FLS devices running software release 04.2.00 and later (AES)

3DES and AES are the encryption algorithms supported in Foundry’s implementation of SSHv2.

Supported MAC (Message Authentication Code) Algorithm

SHA 1 is the MAC algorithm supported in Foundry’s implementation of SSHv2.

AES Encryption for SSHv2

Platform and Software Support:

- FGS and FLS devices running software release 04.2.00 and later
- FESX/FSX/FWSX devices running software release 04.1.00 and later – L2, BL3, L3

Foundry SSHv2 (Secure Shell version 2 protocol) uses the Allegro RomPager4.61 ROMSShell toolkit to provide an SSH server. The SSH server allows secure remote access management functions on a Foundry device. SSH provides a function that is similar to Telnet, but unlike Telnet, SSH provides a secure, encrypted connection. The Foundry SSHv2 supports the Advanced Encryption Standard (AES) data encryption, as well as the Data Encryption Standard (DES).

AES has been adopted by the U.S. Government as an encryption standard. The following AES features are supported by FastIron devices using the Foundry SSHv2 protocol:

- diffie-hellman-group 1-sha1 and difie-hellman-group 14-sha1 for key exchange
- SSH-DSS public key algorithm
- 3des-cbc, aes128-cbc, aes192-cbc or aes256-cbc encryption
- hmac-sha1 ensures data integrity
- Password and public key authentication

- TCP/IP forwarding
- X11 forwarding

A total of five SSH connections can be active on a Foundry device. To display information about SSH connections, enter the following command:

```
FastIron#show ip ssh

Connection    Version    Encryption    Username
     1         SSH-2      3des-cbc     Raymond
     2         SSH-2      3des-cbc     Ron
     3         SSH-2      aes128-cbc   David
     4         SSH-2      aes192-cbc   Francesca
     5         SSH-2      aes256-cbc   Bob
```

You can also use the **show who** command to display information about SSH connections:

```
FastIron#show who
       Console connections:
       Established
       you are connecting to this session
       2 minutes 56 seconds in idle

SSH connections:
1. established, client ip address 2.2.2.1, user is Raymond
   1 minutes 15 seconds in idle
2. established, client ip addres 2.2.2.2, user is Ron
   2 minutes 25 seconds in idle
3. established, client ip address 2.2.2.1, user is David
   1 minutes 8 seconds in idle
4. established, client ip address 2.2.2.1, user is Franchesca
   2 minutes 32 seconds in idle
5. established, client ip address 2.2.2.3, user is Bob
   5 minutes 17 seconds in idle
```

To terminate an active connection, enter the following command:

```
FastIron#kill ssh 1
```

Syntax: kill ssh <connection-id>

Configuring SSHv2

Foundry's implementation of SSHv2 supports two kinds of user authentication:

- ***DSA challenge-response authentication***, where a collection of public keys are stored on the device. Only clients with a private key that corresponds to one of the stored public keys can gain access to the device using SSH.

  NOTE: SSHv2 supports and validates DSA keys only. It does not support or validate SSHv1 RSA keys.

- ***Password authentication***, where users attempting to gain access to the device using an SSH client are authenticated with passwords stored on the device or on a TACACS/TACACS+ or RADIUS server

Both kinds of user authentication are enabled by default. You can configure the device to use one or both of them.

To configure Secure Shell on a Foundry device, do the following:

1. If necessary, recreate the SSH keys
2. Generate a host DSA public and private key pair for the device
3. Configure DSA challenge-response authentication
4. Set optional parameters

You can also view information about active SSH connections on the device as well as terminate them.

Recreating SSH Keys

You must recreate SSH keys after any one of the following events:

- after upgrading to FSX software release 03.0.00, which supports SSHv2.
- after downgrading from FSX software release 03.0.00. Releases prior to 03.0.00 support SSHv1.
- after downgrading from FSX software release 04.1.00 to a previous release.
- after upgrading from FGS software release 02.4.00 to release 02.5.00, which supports SSHv2.
- after downgrading from FGS software release 02.5.00. Releases prior to 02.5.00 support SSHv1.

To recreate SSH keys, enter the following command:

```
FastIron(config)#crypto key generate
```

Syntax: crypto key generate

Generating a Host Key Pair

When SSH is configured, a public and private ***host DSA key pair*** is generated for the Foundry device. The SSH server on the Foundry device uses this host DSA key pair, along with a dynamically generated ***server DSA key pair***, to negotiate a session key and encryption method with the client trying to connect to it.

The host DSA key pair is stored in the Foundry device's system-config file. Only the public key is readable. The public key should be added to a "known hosts" file (for example, $HOME/.ssh/known_hosts on UNIX systems) on the clients who want to access the device. Some SSH client programs add the public key to the known hosts file automatically; in other cases, you must manually create a known hosts file and place the Foundry device's public key in it.

While the SSH listener exists at all times, sessions can't be started from clients until a key is generated. Once a key is generated, clients can start sessions. The keys are also not displayed in the configuration file by default. To display the keys, use the **ssh show-host-keys** command in Privileged EXEC mode. To generate a public and private DSA host key pair on a Foundry device, enter the following commands:

```
FastIron(config)#crypto key generate
```

When a host key pair is generated, it is saved to the flash memory of all management modules.

To disable SSHv2 on a Foundry device, enter the following commands:

```
FastIron(config)#crypto key zeroize
```

When SSH is disabled, it is deleted from the flash memory of all management modules.

Syntax: crypto key generate | zeroize

The **generate** keyword places an DSA host key pair in the flash memory and enables SSH on the device.

The **zeroize** keyword deletes the DSA host key pair from the flash memory and disables SSH on the device.

By default, public keys are hidden in the running configuration. You can optionally configure the Foundry device to display the DSA host key pair in the running configuration file entering the following command:

```
FastIron#ssh show-host-keys
```

Syntax: ssh show-host-keys

To hide the public keys in the running configuration file, enter the following command:

```
FastIron#ssh no-show-host-keys
```

Syntax: ssh no-show-host-keys

Providing the Public Key to Clients

If you are using SSH to connect to a Foundry device from a UNIX system, you may need to add the Foundry device's public key to a "known hosts" file; for example, $HOME/.ssh/known_hosts. The following is an example of an entry in a known hosts file:

```
AAAAB3NzaC1kc3MAAACBAPY8ZOHY2yFSJA6XYC9HRwNHxaehvx5wOJ0rzZdzoSOXxbET W6ToHv8D1UJ/
z+zHo9Fiko5XybZnDIaBDHtblQ+Yp7StxyltHnXF1YLfKD1G4T6JYrdH YI14Om
1eg9e4NnCRleaqoZPF3UGfZia6bXrGTQf3gJq2e7Yisk/gF+1VAAAAFQDb8D5cv
wHWTZDPfX0D2s9Rd7NBvQAAAIEAlN92+Bb7D4KLYk3IwRbXblwXdkPggA4pfdtW9v
GfJ0/RHd+NjB4eo1D+0dix6tXwYGN7PKS5R/FXPNwxHPapcj9uL1Jn2AWQ2dsknf+i/FAA
vioUPkmdMc0zuWoSOEsSNhVDtX3WdvVcGcBq9cetzrtOKWOocJmJ80qadxTRHtUAAACB
AN7CY+KKv1gHpRzFwdQm7HK9bb1LAo2KwaoXnadFgeptNBQeSXG1vO+JsvphVMBJc9HS
n24VYtYtsMu74qXviYjziVucWKjjKEb11juqnF0GDlB3VVmxHLmxnAz643WK42Z7dLM5
sY29ouezv4Xz2PuMch5VGPP+CDqzCM4loWgV
```

Configuring DSA Challenge-Response Authentication

With DSA challenge-response authentication, a collection of clients' public keys are stored on the Foundry device. Clients are authenticated using these stored public keys. Only clients that have a private key that corresponds to one of the stored public keys can gain access to the device using SSH.

When DSA challenge-response authentication is enabled, the following events occur when a client attempts to gain access to the device using SSH:

1. The client sends its public key to the Foundry device.
2. The Foundry device compares the client's public key to those stored in memory.
3. If there is a match, the Foundry device uses the public key to encrypt a random sequence of bytes.
4. The Foundry device sends these encrypted bytes to the client.
5. The client uses its private key to decrypt the bytes.
6. The client sends the decrypted bytes back to the Foundry device.
7. The Foundry device compares the decrypted bytes to the original bytes it sent to the client. If the two sets of bytes match, it means that the client's private key corresponds to an authorized public key, and the client is authenticated.

Setting up DSA challenge-response authentication consists of the following steps:

1. Importing authorized public keys into the Foundry device.
2. Enabling DSA challenge response authentication

Importing Authorized Public Keys into the Foundry Device

SSH clients that support DSA authentication normally provide a utility to generate an DSA key pair. The private key is usually stored in a password-protected file on the local host; the public key is stored in another file and is not protected. You should collect one public key from each client to be granted access to the Foundry device and place all of these keys into one file. This public key file is imported into the Foundry device.

The following is an example of a public key file containing one public key:

```
---- BEGIN SSH2 PUBLIC KEY ----
Comment: DSA Public Key
AAAAB3NzaC1kc3MAAACBAPY8ZOHY2yFSJA6XYC9HRwNHxaehvx5wOJ0rzZdzoSOXxbET W6ToHv8D1UJ/
z+zHo9Fiko5XybZnDIaBDHtblQ+Yp7StxyltHnXF1YLfKD1G4T6JYrdH YI14Om
1eg9e4NnCRleaqoZPF3UGfZia6bXrGTQf3gJq2e7Yisk/gF+1VAAAAFQDb8D5cv
wHWTZDPfX0D2s9Rd7NBvQAAAIEAlN92+Bb7D4KLYk3IwRbXblwXdkPggA4pfdtW9v
GfJ0/RHd+NjB4eo1D+0dix6tXwYGN7PKS5R/FXPNwxHPapcj9uL1Jn2AWQ2dsknf+i/FAA
vioUPkmdMc0zuWoSOEsSNhVDtX3WdvVcGcBq9cetzrtOKWOocJmJ80qadxTRHtUAAACB
AN7CY+KKv1gHpRzFwdQm7HK9bb1LAo2KwaoXnadFgeptNBQeSXG1vO+JsvphVMBJc9HS
n24VYtYtsMu74qXviYjziVucWKjjKEb11juqnF0GDlB3VVmxHLmxnAz643WK42Z7dLM5
sY29ouezv4Xz2PuMch5VGPP+CDqzCM4loWgV
---- END SSH2 PUBLIC KEY ----
```

You can import the authorized public keys into the active configuration by loading them from a file on a TFTP server. If you import a public key file from a TFTP server, the file is automatically loaded into the active configuration the next time the device is booted.

To cause a public key file called pkeys.txt to be loaded from a TFTP server each time the Foundry device is booted, enter a command such as the following:

```
FastIron(config)#ip ssh pub-key-file tftp 192.168.1.234 pkeys.txt
```

Syntax: ip ssh pub-key-file tftp | <tftp-server-ip-addr> <filename> [remove]

The <tftp-server-ip-addr> variable is the IP address of the tftp server that contains the public key file that you want to import into the Foundry device.

The <filename> variable is the name of the dsa public key file that you want to import into the Foundry device.

The **remove** parameter deletes the key from the system.

To display the currently loaded public keys, enter the following command:

```
FastIron#show ip client-pub-key

---- BEGIN SSH2 PUBLIC KEY ----
Comment: DSA Public Key
AAAAB3NzaC1kc3MAAACBAPY8ZOHY2yFSJA6XYC9HRwNHxaehvx5wOJ0rzZdzoSOXxbET W6ToHv8D1UJ/
z+zHo9Fiko5XybZnDIaBDHtblQ+Yp7StxyltHnXF1YLfKD1G4T6JYrdH YI14Om
1eg9e4NnCRleaqoZPF3UGfZia6bXrGTQf3gJq2e7Yisk/gF+1VAAAAFQDb8D5cv
wHWTZDPfX0D2s9Rd7NBvQAAAIEAlN92+Bb7D4KLYk3IwRbXblwXdkPggA4pfdtW9v
GfJ0/RHd+NjB4eo1D+0dix6tXwYGN7PKS5R/FXPNwxHPapcj9uL1Jn2AWQ2dsknf+i/FAA
vioUPkmdMc0zuWoSOEsSNhVDtX3WdvVcGcBq9cetzrtOKWOocJmJ80qadxTRHtUAAACB
AN7CY+KKv1gHpRzFwdQm7HK9bb1LAo2KwaoXnadFgeptNBQeSXG1vO+JsvphVMBJc9HS
n24VYtYtsMu74qXviYjziVucWKjjKEb11juqnF0GDlB3VVmxHLmxnAz643WK42Z7dLM5
sY29ouezv4Xz2PuMch5VGPP+CDqzCM4loWgV
---- END SSH2 PUBLIC KEY ----
```

Syntax: show ip client-pub-key [| begin<expression> | exclude <expression> | include <expression>]

To clear the public keys from the buffers, enter the following command:

```
FastIron#clear public-key
```

Syntax: clear public-key

Use the **ip ssh pub-key remove** command to delete the public key from the system.

Enabling DSA Challenge-Response Authentication

DSA challenge-response authentication is enabled by default. You can disable or re-enable it manually.

To enable DSA challenge-response authentication:

```
FastIron(config)#ip ssh key-authentication yes
```

To disable DSA challenge-response authentication:

```
FastIron(config)#ip ssh key-authentication no
```

Syntax: ip ssh key-authentication yes | no

Setting Optional Parameters

You can adjust the following SSH settings on the Foundry device:

- The number of SSH authentication retries
- The user authentication method the Foundry device uses for SSH connections
- Whether the Foundry device allows users to log in without supplying a password
- The port number for SSH connections
- The SSH login timeout value
- A specific interface to be used as the source for all SSH traffic from the device
- The maximum idle time for SSH sessions

Setting the Number of SSH Authentication Retries

By default, the Foundry device attempts to negotiate a connection with the connecting host three times. The number of authentication retries can be changed to between 1 – 5.

For example, the following command changes the number of authentication retries to 5:

```
FastIron(config)#ip ssh authentication-retries 5
```

Syntax: ip ssh authentication-retries <number>

Deactivating User Authentication

After the SSH server on the Foundry device negotiates a session key and encryption method with the connecting client, user authentication takes place. Foundry’s implementation of SSH supports DSA challenge-response authentication and password authentication.

With DSA challenge-response authentication, a collection of clients’ public keys are stored on the Foundry device. Clients are authenticated using these stored public keys. Only clients that have a private key that corresponds to one of the stored public keys can gain access to the device using SSH.

With password authentication, users are prompted for a password when they attempt to log into the device (provided empty password logins are not allowed). If there is no user account that matches the user name and password supplied by the user, the user is not granted access.

You can deactivate one or both user authentication methods for SSH. Note that deactivating both authentication methods essentially disables the SSH server entirely.

To disable DSA challenge-response authentication:

```
FastIron(config)#ip ssh key-authentication no
```

Syntax: ip ssh key-authentication yes | no

The default is “yes”.

To deactivate password authentication:

```
FastIron(config)#ip ssh password-authentication no
```

Syntax: ip ssh password-authentication no | yes

The default is “yes”.

Enabling Empty Password Logins

By default, empty password logins are not allowed. This means that users with an SSH client are always prompted for a password when they log into the device. To gain access to the device, each user must have a user name and password. Without a user name and password, a user is not granted access.

If you enable empty password logins, users are ***not*** prompted for a password when they log in. Any user with an SSH client can log in without being prompted for a password.

To enable empty password logins:

```
FastIron(config)#ip ssh permit-empty-passwd yes
```

Syntax: ip ssh permit-empty-passwd no | yes

Setting the SSH Port Number

By default, SSH traffic occurs on TCP port 22. You can change this port number. For example, the following command changes the SSH port number to 2200:

```
FastIron(config)#ip ssh port 2200
```

Note that if you change the default SSH port number, you must configure SSH clients to connect to the new port. Also, you should be careful not to assign SSH to a port that is used by another service. If you change the SSH port number, Foundry recommends that you change it to a port number greater than 1024.

Syntax: ip ssh port <number>

Setting the SSH Login Timeout Value

When the SSH server attempts to negotiate a session key and encryption method with a connecting client, it waits a maximum of 120 seconds for a response from the client. If there is no response from the client after 120 seconds, the SSH server disconnects. You can change this timeout value to between 1 – 120 seconds. For example, to change the timeout value to 60 seconds:

```
FastIron(config)#ip ssh timeout 60
```

Syntax: ip ssh timeout <seconds>

Designating an Interface as the Source for All SSH Packets (Layer 3 Code Only)

You can designate a loopback interface, virtual interface, or Ethernet port as the source for all SSH packets from the device. The software uses the IP address with the numerically lowest value configured on the port or interface as the source IP address for SSH packets originated by the device.

NOTE: When you specify a single SSH source, you can use only that source address to establish SSH management sessions with the Foundry device.

To specify the numerically lowest IP address configured on a loopback interface as the device’s source for all SSH packets, enter commands such as a the following:

```
FastIron(config)#int loopback 2
FastIron(config-lbif-2)#ip address 10.0.0.2/24
FastIron(config-lbif-2)#exit
FastIron(config)#ip ssh source-interface loopback 2
```

The commands in this example configure loopback interface 2, assign IP address 10.0.0.2/24 to the interface, then designate the interface as the source for all SSH packets from the Foundry device.

Syntax: ip ssh source-interface ethernet <slot/port> | loopback <num> | ve <num>

The <num> parameter is a loopback interface or virtual interface number. The <slot/port> parameter specifies an ethernet port number. For example:

```
FastIron(config)#interface ethernet 1/4
FastIron(config-if-e10000-1/4)#ip address 209.157.22.110/24
FastIron(config-if-e10000-1/4)#exit
FastIron(config)#ip ssh source-interface ethernet 1/4
```

Configuring the Maximum Idle Time for SSH Sessions

By default, SSH sessions do not time out. Optionally, you can set the amount of time an SSH session can be inactive before the Foundry device closes it. For example, to set the maximum idle time for SSH sessions to 30 minutes:

```
FastIron(config)#ip ssh idle-time 30
```

Syntax: ip ssh idle-time <minutes>

If an established SSH session has no activity for the specified number of minutes, the Foundry device closes it. An idle time of 0 minutes (the default value) means that SSH sessions never time out. The maximum idle time for SSH sessions is 240 minutes.

Filtering SSH Access Using ACLs

You can permit or deny SSH access to the Foundry device using ACLs. To use ACLs, first create the ACLs you want to use. You can specify a numbered standard IPv4 ACL, a named standard IPv4 ACL

Enter commands such as the following:

```
FastIron(config)#access-list 10 permit host 192.168.144.241
FastIron(config)#access-list 10 deny host 192.168.144.242 log
FastIron(config)#access-list 10 permit host 192.168.144.243
FastIron(config)#access-list 10 deny any
FastIron(config)#ssh access-group 10
```

Syntax: ssh access-group <standard-named-acl> | <standard-numbered-acl>

Terminating an Active SSH Connection

To terminate one of the active SSH connections, enter the following command:

```
FastIron#kill ssh 1
```

Syntax: kill ssh <connection-id>

Displaying SSH Connection Information

Up to five SSH connections can be active on the Foundry device. To display information about SSH connections, enter the following command:

```
FastIron#show ip ssh
Connection Version  Encryption  Username
1           SSH-2   3des-cbc    Hanuma
2           SSH-2   3des-cbc    Mikaila
3           SSH-2   3des-cbc    Jenny
4           SSH-2   3des-cbc    Mariah
5           SSH-2   3des-cbc    Logan
```

Syntax: show ip ssh [| begin <expression> | exclude <expression> | include <expression>]

This display shows the following information about the active SSH connections:

Table 41.1: SSH Connection Information

This Field...	Displays...
Connection	The SSH connection ID. This can be from 1 – 5.
Version	The SSH version number. This should always be 1.5.
Encryption	The encryption method used for the connection.
Username	The user name for the connection.

The **show who** command also displays information about SSH connections. For example:

```
FastIron#show who
Console connections:
established, monitor enabled, in config mode
2 minutes 17 seconds in idle
Telnet connections (inbound):
1 closed
2 closed
3 closed
4 closed
5 closed
Telnet connection (outbound):
6 closed
SSH connections:
1 established, client ip address 192.168.144.241, user is hanuma
1 minutes 16 seconds in idle
2 established, client ip address 192.168.144.241, user is Mikaila
you are connecting to this session
18 seconds in idle
3 established, client ip address 192.168.144.241, user is Jenny
1 minutes 39 seconds in idle
4 established, client ip address 192.168.144.242, user is Mariah
41 seconds in idle
5 established, client ip address 192.168.144.241, user is Logan
23 seconds in idle
```

Syntax: show who [| begin<expression> | exclude<expression> | include<expression>]

Using Secure Copy with SSHv2

Secure Copy (SCP) uses security built into SSH to transfer image and configuration files to and from the device. SCP automatically uses the authentication methods, encryption algorithm, and data compression level configured for SSH. For example, if password authentication is enabled for SSH, the user is prompted for a user name and password before SCP allows a file to be transferred. No additional configuration is required for SCP on top of SSH.

You can use SCP to copy files on the Foundry device, including the startup configuration and running configuration files, to or from an SCP-enabled remote host.

Enabling and Disabling SCP

SCP is enabled by default and can be disabled. To disable SCP, enter the following command:

```
FastIron(config)#ip ssh scp disable
```

Syntax: ip ssh scp disable | enable

NOTE: If you disable SSH, SCP is also disabled.

NOTE: When using SCP, you enter the **scp** commands on the SCP-enabled client, rather than the console on the Foundry device.

NOTE: Certain SCP client options, including -p and -r, are ignored by the SCP server on the Foundry device. If an option is ignored, the client is notified.

Example File Transfers using SCP

The following are examples of using SCP to transfer files to and from a Foundry device.

Copying a File to the Running Config

To copy a configuration file (c:\cfg\foundry.cfg) to the running configuration file on a Foundry device at 192.168.1.50 and log in as user terry, enter the following command on the SCP-enabled client:

```
C:\> scp c:\cfg\foundry.cfg terry@192.168.1.50:runConfig
```

If password authentication is enabled for SSH, the user is prompted for user terry's password before the file transfer takes place.

Copying a File to the Startup Config

To copy the configuration file to the startup configuration file:

```
C:\> scp c:\cfg\foundry.cfg terry@192.168.1.50:startConfig
```

Copying the Running Config File to an SCP-Enabled Client

To copy the running configuration file on the Foundry device to a file called c:\cfg\fdryrun.cfg on the SCP-enabled client:

```
C:\> scp terry@192.168.1.50:runConfig c:\cfg\fdryrun.cfg
```

Copying the Startup Config File to an SCP-Enabled Client

To copy the startup configuration file on the Foundry device to a file called c:\cfg\fdrystart.cfg on the SCP-enabled client:

```
C:\> scp terry@192.168.1.50:startConfig c:\cfg\fdrystart.cfg
```

To overwrite the running configuration file:

```
C:\> scp c:\cfg\foundry.cfg terry@192.168.1.50:runConfig-overwrite
```

Copying a Software Image File to Flash Memory

Platform Support:

- FESX/FSX/FWSX devices running software release 03.0.00 and later

To copy a software image file from an SCP-enabled client to the primary flash on the Foundry device:

```
C:\> scp SXL03200.bin terry@192.168.1.50:flash:pri:SXL03200.bin
```

or

```
C:\> scp SXL03200.bin terry@192.168.1.50:flash:primary.bin
```

To copy a software image file from an SCP-enabled client to the secondary flash on the Foundry device:

```
C:\> scp SXL03200.bin terry@192.168.1.50:flash:sec:SXL03200.bin
```

or

```
C:\> scp SXL03200.bin terry@192.168.1.50:flash:secondary.bin
```

NOTE: The Foundry device supports only one SCP copy session at a time.

Copying a Software Image File from Flash Memory

Platform Support:

- FESX/FSX/FWSX devices running software release 03.0.00 and later

To copy a software image from the primary flash on the Foundry device to an SCP-enabled client:

```
C:\ scp terry@192.168.1.50:flash:primary.bin SXL03200.bin
```

To copy a software image from the secondary flash on the Foundry device to an SCP-enabled client:

```
C:\ scp terry@192.168.1.50:flash:secondary.bin SXL03200.bin
```

NOTE: The Foundry device supports only one SCP copy session at a time.

Chapter 42

Configuring 802.1X Port Security

Foundry devices support the IEEE 802.1X standard for authenticating devices attached to LAN ports. Using 802.1X port security, you can configure a Foundry device to grant access to a port based on information supplied by a client to an authentication server.

When a user logs on to a network that uses 802.1X port security, the Foundry device grants (or doesn't grant) access to network services after the user is authenticated by an authentication server. The user-based authentication in 802.1X port security provides an alternative to granting network access based on a user's IP address, MAC address, or subnetwork.

IETF RFC Support

Foundry's implementation of 802.1X port security supports the following RFCs:

- RFC 2284 PPP Extensible Authentication Protocol (EAP)
- RFC 2865 Remote Authentication Dial In User Service (RADIUS)
- RFC 2869 RADIUS Extensions

How 802.1X Port Security Works

This section explains the basic concepts behind 802.1X port security, including device roles, how the devices communicate, and the procedure used for authenticating clients.

NOTE: 802.1X Port Security cannot be configured on MAC Port Security-enabled ports.

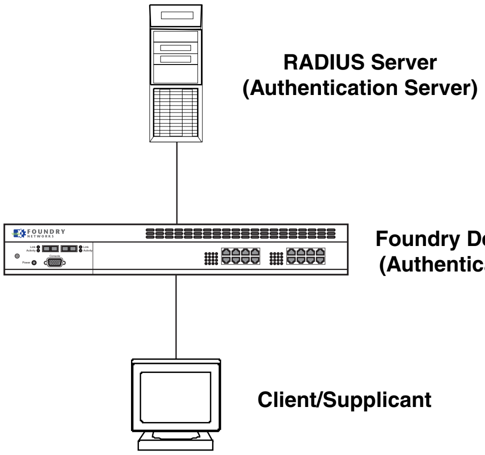
Device Roles in an 802.1X Configuration

The 802.1X standard defines the roles of ***Client/Supplicant***, ***Authenticator***, and ***Authentication Server*** in a network.

The Client (known as a ***Supplicant*** in the 802.1X standard) provides username/password information to the Authenticator. The Authenticator sends this information to the Authentication Server. Based on the Client's information, the Authentication Server determines whether the Client can use services provided by the Authenticator. The Authentication Server passes this information to the Authenticator, which then provides services to the Client, based on the authentication result.

Figure 42.1 illustrates these roles.

Figure 42.1 Authenticator, Client/Supplicant, and Authentication Server in an 802.1X Configuration



Authenticator – The device that controls access to the network. In an 802.1X configuration, the Foundry device serves as the Authenticator. The Authenticator passes messages between the Client and the Authentication Server. Based on the identity information supplied by the Client, and the authentication information supplied by the Authentication Server, the Authenticator either grants or does not grant network access to the Client.

Client/Supplicant – The device that seeks to gain access to the network. Clients must be running software that supports the 802.1X standard (for example, the Windows XP operating system). Clients can either be directly connected to a port on the Authenticator, or can be connected by way of a hub.

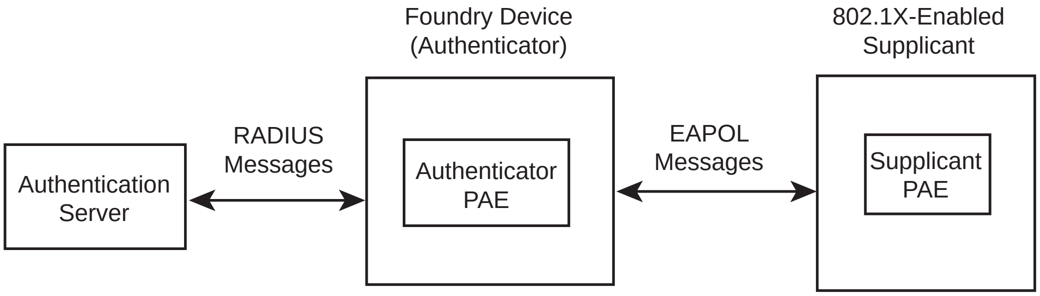
Authentication Server – The device that validates the Client and specifies whether or not the Client may access services on the device. Foundry supports Authentication Servers running RADIUS.

Communication between the Devices

For communication between the devices, 802.1X port security uses the ***Extensible Authentication Protocol*** (EAP), defined in RFC 2284. The 802.1X standard specifies a method for encapsulating EAP messages so that they can be carried over a LAN. This encapsulated form of EAP is known as EAP over LAN (***EAPOL***). The standard also specifies a means of transferring the EAPOL information between the Client/Supplicant, Authenticator, and Authentication Server.

EAPOL messages are passed between the ***Port Access Entity (PAE)*** on the Supplicant and the Authenticator. Figure 42.2 shows the relationship between the Authenticator PAE and the Supplicant PAE.

Figure 42.2 Authenticator PAE and Supplicant PAE



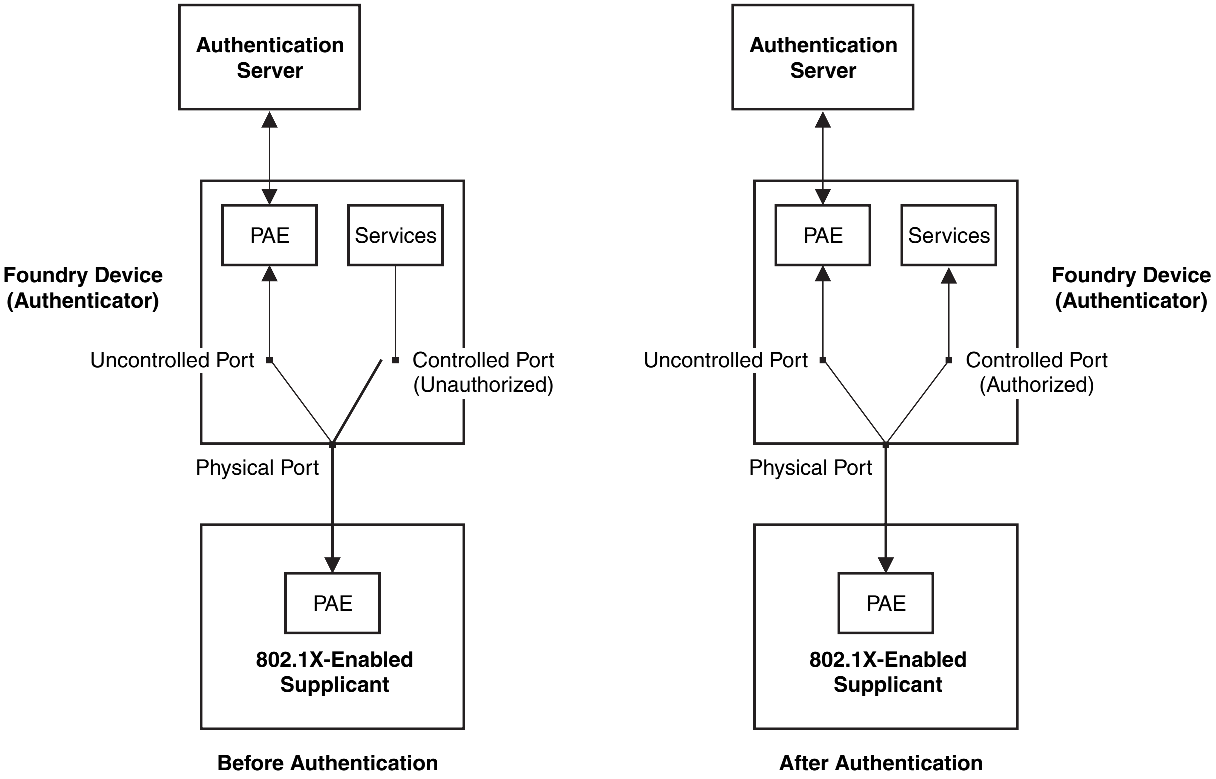
Authenticator PAE – The Authenticator PAE communicates with the Supplicant PAE, receiving identifying information from the Supplicant. Acting as a RADIUS client, the Authenticator PAE passes the Supplicant’s information to the Authentication Server, which decides whether the Supplicant can gain access to the port. If the Supplicant passes authentication, the Authenticator PAE grants it access to the port.

Supplicant PAE – The Supplicant PAE supplies information about the Client to the Authenticator PAE and responds to requests from the Authenticator PAE. The Supplicant PAE can also initiate the authentication procedure with the Authenticator PAE, as well as send log off messages.

Controlled and Uncontrolled Ports

A physical port on the device used with 802.1X port security has two virtual access points: a ***controlled*** port and an ***uncontrolled*** port. The controlled port provides full access to the network. The uncontrolled port provides access only for EAPOL traffic between the Client and the Authentication Server. When a Client is successfully authenticated, the controlled port is opened to the Client. Figure 42.3 illustrates this concept.

Figure 42.3 Controlled and Uncontrolled Ports Before and After Client Authentication



Before a Client is authenticated, only the uncontrolled port on the Authenticator is open. The uncontrolled port allows only EAPOL frames to be exchanged between the Client and the Authentication Server. The controlled port is in the unauthorized state and allows no traffic to pass through.

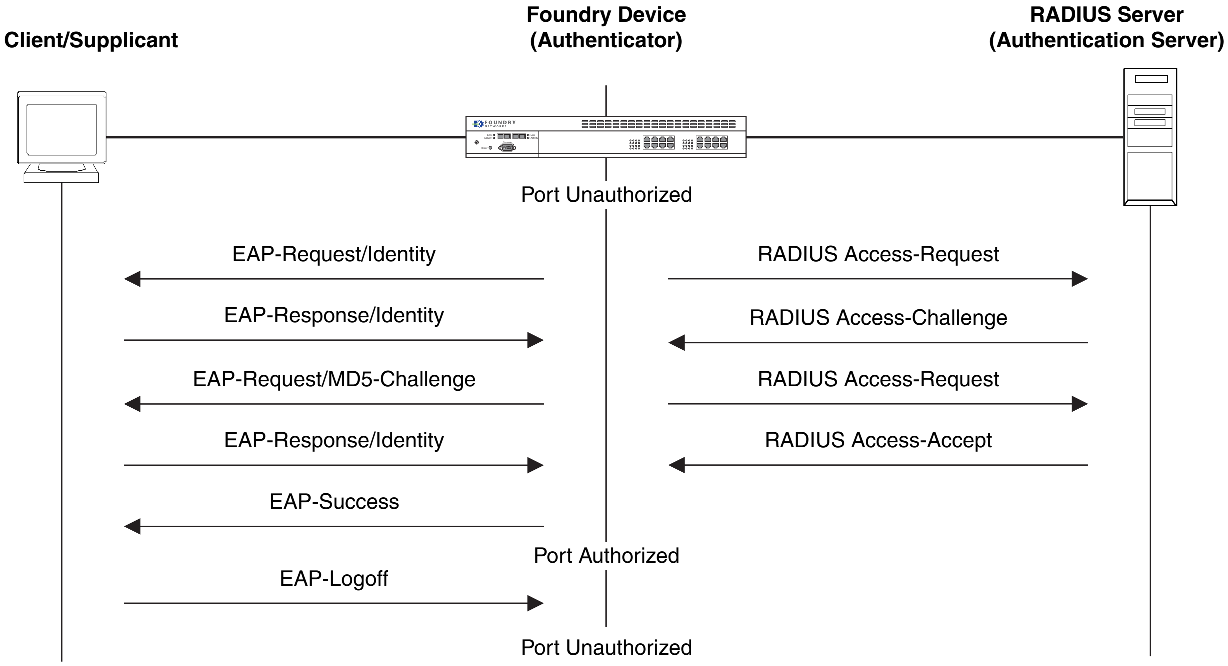
During authentication, EAPOL messages are exchanged between the Supplicant PAE and the Authenticator PAE, and RADIUS messages are exchanged between the Authenticator PAE and the Authentication Server. See “Message Exchange During Authentication” on page 42-4 for an example of this process. If the Client is successfully authenticated, the controlled port becomes authorized, and traffic from the Client can flow through the port normally.

By default, all controlled ports on the Foundry device are placed in the authorized state, allowing all traffic. When authentication is activated on an 802.1X-enabled interface, the interface’s controlled port is placed initially in the unauthorized state. When a Client connected to the port is successfully authenticated, the controlled port is then placed in the authorized state until the Client logs off. See “Enabling 802.1X Port Security” on page 42-17 for more information.

Message Exchange During Authentication

Figure 42.4 illustrates a sample exchange of messages between an 802.1X-enabled Client, a Foundry device acting as Authenticator, and a RADIUS server acting as an Authentication Server.

Figure 42.4 Message Exchange between Client/Supplicant, Authenticator, and Authentication Server



In this example, the Authenticator (the Foundry device) initiates communication with an 802.1X-enabled Client. When the Client responds, it is prompted for a username (255 characters maximum) and password. The Authenticator passes this information to the Authentication Server, which determines whether the Client can access services provided by the Authenticator. When the Client is successfully authenticated by the RADIUS server, the port is authorized. When the Client logs off, the port becomes unauthorized again.

Foundry's 802.1X implementation supports dynamic VLAN assignment. If one of the attributes in the Access-Accept message sent by the RADIUS server specifies a VLAN identifier, and this VLAN is available on the Foundry device, the client's port is moved from its default VLAN to the specified VLAN. When the client disconnects from the network, the port is placed back in its default VLAN. See "Configuring Dynamic VLAN Assignment for 802.1X Ports" on page 42-11 for more information.

If a Client does not support 802.1X, authentication cannot take place. The Foundry device sends EAP-Request/Identity frames to the Client, but the Client does not respond to them.

When a Client that supports 802.1X attempts to gain access through a non-802.1X-enabled port, it sends an EAP start frame to the Foundry device. When the device does not respond, the Client considers the port to be authorized, and starts sending normal traffic.

Foundry devices support Identity and MD5-challenge requests in EAP Request/Response messages. However, FESX release 02.1.01 and FSX release 02.3.01 support the following 802.1X authentication challenge types:

NOTE: See also "EAP Pass-Through Support" on page 42-5.

- EAP-TLS (RFC 2716) – EAP Transport Level Security (TLS) provides strong security by requiring both client and authentication server to be identified and validated through the use of public key infrastructure (PKI) digital certificates. EAP-TLS establishes a tunnel between the client and the authentication server to protect messages from unauthorized users' eavesdropping activities. Since EAP-TLS requires PKI digital certificates on both the clients and the authentication servers, the roll out, maintenance, and scalability of this authentication method is much more complex than other methods. EAP-TLS is best for installations with existing PKI certificate infrastructures.

- EAP-TTLS (Internet-Draft) – The EAP Tunnelled Transport Level Security (TTLS) is an extension of EAP-TLS Like TLS, EAP-TTLS provides strong authentication; however it requires only the authentication server to be validated by the client through a certificate exchange between the server and the client. Clients are authenticated by the authentication server using user names and passwords.

  A TLS tunnel can be used to protect EAP messages and existing user credential services such as Active Directory, RADIUS, and LDAP. Backward compatibility for other authentication protocols such as PAP, CHAP, MS-CHAP, and MS-CHAP-V2 are also provided by EAP-TTLS. EAP-TTLS is not considered foolproof and can be fooled into sending identity credentials if TLS tunnels are not used. EAP-TTLS is suited for installations that require strong authentication without the use of mutual PKI digital certificates.

- PEAP (Internet-Draft) – Protected EAP Protocol (PEAP) is an Internet-Draft that is similar to EAP-TTLS. PEAP client authenticates directly with the backend authentication server. The authenticator acts as a pass-through device, which does not need to understand the specific EAP authentication protocols.

  Unlike EAP-TTLS, PEAP does not natively support user name and password to authenticate clients against an existing user database such as LDAP. PEAP secures the transmission between the client and authentication server with a TLS encrypted tunnel. PEAP also allows other EAP authentication protocols to be used. It relies on the mature TLS keying method for its key creation and exchange. PEAP is best suited for installations that require strong authentication without the use of mutual certificates.

NOTE: If the 802.1X Client will be sending a packet that is larger than 1500 bytes, you must enable **jumbo** at the Global config level of the CLI.

Configuration for these challenge types is the same as for the EAP-MD5 challenge type.

EAP Pass-Through Support

Platform Support:

- FESX/FSX/FWSX devices running software release 02.3.01 and later

EAP pass-through support is fully compliant with RFC 3748, in which, by default, compliant pass-through authenticator implementations forward EAP challenge request packets of any type. Previous software releases were limited to MD5, EAP-TTLS, EAP-PEAP, and EAP-TLS challenge request types for 802.1X authentication.

NOTE: If the 802.1X supplicant or authentication server will be sending packets that are greater than 1500 MTU, you should configure the device to accommodate a bigger buffer size.

Support for RADIUS User-name Attribute in Access-Accept Messages

Platform Support:

- FESX/FSX/FWSX devices running software release 04.0.00 and later
- FGS and FLS devices running software release 04.0.00 and later

Foundry 802.1X-enabled ports support the RADIUS User-name (type 1) attribute in the Access-Accept message returned during 802.1X authentication.

This feature is useful when the client/supplicant doesn't provide its user-name in the EAP-response/identity frame, and the username is key to providing useful information. For example, when the User-name attribute is sent in the Access-Accept message, it is then available for display in sFlow sample messages sent to a collector, and in the output of some **show dot1x** CLI commands, such as **show dot1x mac-sessions**.

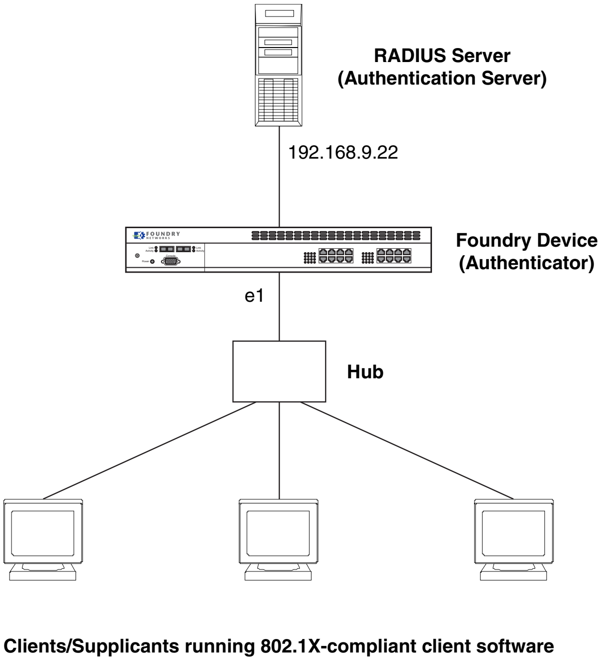
To enable this feature, add the following attribute on the RADIUS server:

Attribute Name	Type	Value
User-name	1	<name> (string)

Authenticating Multiple Hosts Connected to the Same Port

Foundry devices support 802.1X authentication for ports with more than one host connected to them. Figure 42.5 illustrates a sample configuration where multiple hosts are connected to a single 802.1X port.

Figure 42.5 Multiple Hosts Connected to a Single 802.1X-Enabled Port



The way the Foundry device authenticates Clients in a multiple-host configuration depends on the software release running on the device:

Starting in release 02.2.00 for the FESX and FWSX and release 02.3.01 for the FSX, if there are multiple hosts connected to a single 802.1X-enabled port, the Foundry device authenticates each of them individually. Each host's authentication status is independent of the others, so that if one authenticated host disconnects from the network, it has no effect on the authentication status of any of the other authenticated hosts.

By default, traffic from hosts that cannot be authenticated by the RADIUS server is dropped in hardware. You can optionally configure the Foundry device to assign the port to a "restricted" VLAN if authentication of the Client is unsuccessful.

How 802.1X Multiple-Host Authentication Works

When multiple hosts are connected to a single 802.1X-enabled port on a Foundry device (as in Figure 42.5), 802.1X authentication is performed in the following way:

1. One of the 802.1X-enabled Clients attempts to log into a network in which a Foundry device serves as an Authenticator.
2. The Foundry device creates an internal session (called a ***dot1x-mac-session***) for the Client. A dot1x-mac-session serves to associate a Client's MAC address and username with its authentication status.

3. The Foundry device performs 802.1X authentication for the Client. Messages are exchanged between the Foundry device and the Client, and between the device and the Authentication Server (RADIUS server). The result of this process is that the Client is either successfully authenticated or not authenticated, based on the username and password supplied by the client.
4. If the Client is successfully authenticated, the Client's dot1x-mac-session is set to "access-is-allowed". This means that traffic from the Client can be forwarded normally.
5. If authentication for the Client is unsuccessful the first time, multiple attempts to authenticate the client will be made as determined by the **attempts** variable in the **auth-fail-max-attempts** command.
    - See "Specifying the Number of Authentication Attempts the Device Makes Before Dropping Packets" on page 42-22 for information on how to do this.
6. If authentication for the Client is unsuccessful more than the number of times specified by the **attempts** variable in the **auth-fail-max-attempts** command, an ***authentication-failure action*** is taken. The authentication-failure action can be either to drop traffic from the Client, or to place the port in a "restricted" VLAN.
    - If the authentication-failure action is to drop traffic from the Client, then the Client's dot1x-mac-session is set to "access-denied", causing traffic from the Client to be dropped in hardware.
    - If the authentication-failure action is to place the port in a "restricted" VLAN, If the Client's dot1x-mac-session is set to "access-restricted" then the port is moved to the specified restricted VLAN, and traffic from the Client is forwarded normally.
7. When the Client disconnects from the network, the Foundry device deletes the Client's dot1x-mac-session. This does not affect the dot1x-mac-session or authentication status (if any) of the other hosts connected on the port.

Configuration Notes

- The Client's dot1x-mac-session establishes a relationship between the username and MAC address used for authentication. If a user attempts to gain access from different Clients (with different MAC addresses), he or she would need to be authenticated from each Client.
- If a Client has been denied access to the network (that is, the Client's dot1x-mac-session is set to "access-denied"), then you can cause the Client to be re-authenticated by manually disconnecting the Client from the network, or by using the **clear dot1x mac-session** command. See "Clearing a dot1x-mac-session for a MAC Address" on page 42-23 for information on this command.
- When a Client has been denied access to the network, its dot1x-mac-session is aged out if no traffic is received from the Client's MAC address over a fixed hardware aging period (70 seconds), plus a configurable software aging period. You can optionally change the software aging period for dot1x-mac-sessions or disable aging altogether. After the denied Client's dot1x-mac-session is aged out, traffic from that Client is no longer blocked, and the Client can be re-authenticated.

  In addition, you can configure disable aging for the dot1x-mac-session of Clients that have been granted either full access to the network, or have been placed in a restricted VLAN. After a Client's dot1x-mac-session ages out, the Client must be re-authenticated. See "Disabling Aging for dot1x-mac-sessions" on page 42-22 for more information.
- Dynamic IP ACL and MAC address filter assignment is supported in an 802.1X multiple-host configuration. See "Dynamically Applying IP ACLs and MAC Filters to 802.1X Ports" on page 42-14.
- 802.1X multiple-host authentication has the following additions:
    - Configurable hardware aging period for denied client dot1x-mac-sessions. See "Configurable Hardware Aging Period for Denied Client dot1x-mac-sessions" on page 42-8.
    - Dynamic ACL and MAC address filter assignment in 802.1X multiple-host configurations. See "Dynamically Applying IP ACLs and MAC Filters to 802.1X Ports" on page 42-14.
    - Dynamic multiple VLAN assignment for 802.1X ports. See "Dynamic Multiple VLAN Assignment for 802.1X Ports" on page 42-12.
    - Enhancements to some **show** commands.

- Differences in command syntax for saving dynamic VLAN assignments to the startup-config file.

Configurable Hardware Aging Period for Denied Client dot1x-mac-sessions

When one of the 802.1X-enabled Clients in a multiple-host configuration attempts to log into a network in which a Foundry device serves as an Authenticator, the device creates a dot1x-mac-session for the Client.

When a Client has been denied access to the network, its dot1x-mac-session is aged out if no traffic is received from the Client's MAC address over a period of time. After a denied Client's dot1x-mac-session ages out, the Client can be re-authenticated. Aging of a denied Client's dot1x-mac-session occurs in two phases, known as hardware aging and software aging.

On FastIron X Series devices, the hardware aging period for a denied Client's dot1x-mac-session is not fixed at 70 seconds. The hardware aging period for a denied Client's dot1x-mac-session is equal to the length of time specified with the dot1x **timeout quiet-period** command. By default, the hardware aging time is 60 seconds. Once the hardware aging period ends, the software aging period begins. When the software aging period ends, the denied Client's dot1x-mac-session ages out, and the Client can be authenticated again.

802.1X Port Security and sFlow

sFlow is a system for observing traffic flow patterns and quantities within and among a set of Layer 2 Switches and Layer 3 Switches. sFlow works by taking periodic samples of network data and exporting this information to a collector.

When you enable sFlow forwarding on an 802.1X-enabled interface, the samples taken from the interface include the user name string at the inbound and/or outbound port, if that information is available.

For more information on sFlow, see the chapter “Network Monitoring” on page B-1.

Configuring 802.1X Port Security

Configuring 802.1X port security on a Foundry device consists of the following tasks:

1. Configure the Foundry device's interaction with the Authentication Server:
    - “Configuring an Authentication Method List for 802.1X” on page 42-9
    - “Setting RADIUS Parameters” on page 42-9
    - “Configuring Dynamic VLAN Assignment for 802.1X Ports” on page 42-11 (optional)
    - “Dynamically Applying IP ACLs and MAC Filters to 802.1X Ports” on page 42-14
2. Configure the Foundry device's role as the Authenticator:
    - “Enabling 802.1X Port Security” on page 42-17
    - “Initializing 802.1X on a Port” on page 42-21 (optional)
3. Configure the Foundry device's interaction with Clients:
    - “Configuring Periodic Re-Authentication” on page 42-19 (optional)
    - “Re-Authenticating a Port Manually” on page 42-19 (optional)
    - “Setting the Quiet Period” on page 42-19 (optional)
    - “Setting the Wait Interval for EAP Frame Retransmissions” on page 42-20 (optional)
    - “Setting the Maximum Number of EAP Frame Retransmissions” on page 42-20 (optional)
    - “Specifying a Timeout for Retransmission of Messages to the Authentication Server” on page 42-21 (optional)
    - “Allowing Access to Multiple Hosts” on page 42-21 (optional)
    - “Defining MAC Filters for EAP Frames” on page 42-23 (optional)

Configuring an Authentication Method List for 802.1X

To use 802.1X port security, you must specify an authentication method to be used to authenticate Clients. Foundry supports RADIUS authentication with 802.1X port security. To use RADIUS authentication with 802.1X port security, you create an authentication method list for 802.1X and specify RADIUS as an authentication method, then configure communication between the Foundry device and RADIUS server.

For example:

```
FastIron(config)#aaa authentication dot1x default radius
```

Syntax: [no] aaa authentication dot1x default <method-list>

For the <method-list>, enter at least one of the following authentication methods:

radius – Use the list of all RADIUS servers that support 802.1X for authentication.

none – Use no authentication. The Client is automatically authenticated without the device using information supplied by the Client.

NOTE: If you specify both **radius** and **none**, make sure **radius** comes before **none** in the method list.

Setting RADIUS Parameters

To use a RADIUS server to authenticate access to a Foundry device, you must identify the server to the Foundry device. For example:

```
FastIron(config)#radius-server host 209.157.22.99 auth-port 1812 acct-port 1813
default key mirabeau dot1x
```

Syntax: radius-server host <ip-addr> | <ipv6-addr> | <server-name> [authentication-only | accounting-only | default] [key 0 | 1 <string>] [dot1x]

The **host** <ip-addr> | <ipv6-addr> | <server-name> parameter is either an IP address or an ASCII text string.

The **dot1x** parameter indicates that this RADIUS server supports the 802.1X standard. A RADIUS server that supports the 802.1X standard can also be used to authenticate non-802.1X authentication requests.

NOTE: To implement 802.1X port security, at least one of the RADIUS servers identified to the Foundry device must support the 802.1X standard.

Supported RADIUS Attributes

Many IEEE 802.1X Authenticators will function as RADIUS clients. Some of the RADIUS attributes may be received as part of IEEE 802.1X authentication. Foundry devices support the following RADIUS attributes for IEEE 802.1X authentication:

- Username (1) – RFC 2865
- NAS-IP-Address (4) – RFC 2865
- NAS-Port (5) – RFC 2865
- Service-Type (6) – RFC 2865
- FilterId (11) – RFC 2865
- Framed-MTU (12) – RFC 2865
- State (24) – RFC 2865
- Vendor-Specific (26) – RFC 2865
- Session-Timeout (27) – RFC 2865
- Termination-Action (29) – RFC 2865

- Calling-Station-ID (31) – RFC 2865
- NAS-Port-Type (61) š RFC 2865
- Tunnel-Type (64) – RFC 2868
- Tunnel-Medium-Type (65) – RFC 2868
- EAP Message (79) – RFC 2579
- Message-Authenticator (80) RFC 3579
- Tunnel-Private-Group-Id (81) – RFC 2868
- NAS-Port-id (87) – RFC2869

Specifying the RADIUS Timeout Action

Platform Support:

- FSX devices running software release 03.2.00 and later
- FGS and FLS devices running software release 04.1.00 and later

A RADIUS timeout occurs when the Foundry device does not receive a response from a RADIUS server within a specified time limit and after a certain number of retries. The time limit and number of retries can be manually configured using the CLI commands **radius-server timeout** and **radius-server retransmit**, respectively. If the parameters are not manually configured, the Foundry device applies the default value of three seconds time limit with a maximum of three retries.

In releases prior to FSX software release 03.2.00 and FGS and FLS software release 04.1.00, when a RADIUS timeout occurs, the Foundry device will automatically reset the authentication process then retry to authenticate the user.

In FSX software release 03.2.00, and FGS software release 04.1.00, you can better control port behavior when a RADIUS timeout occurs. That is, you can configure a port on the Foundry device to automatically pass or fail users being authenticated. A ***pass*** essentially bypasses the authentication process and permits user access to the network. A ***fail*** bypasses the authentication process and blocks user access to the network, unless **restrict-vlan** is configured, in which case, the user is placed into a VLAN with restricted or limited access. By default, the Foundry device will reset the authentication process and ***retry*** to authenticate the user.

Specify the RADIUS timeout action at the Interface level of the CLI.

Permit User Access to the Network after a RADIUS Timeout

To set the RADIUS timeout behavior to bypass 802.1X authentication and *permit* user access to the network, enter commands such as the following:

```
FastIron(config)#interface ethernet 3/1
FastIron(config-if-e100-3/1)#dot1x auth-timeout-action success
```

Syntax: [no] dot1x auth-timeout-action success

Once the *success* timeout action is enabled, use the **no** form of the command to reset the RADIUS timeout behavior to *retry*.

Re-authenticate a User

To configure RADIUS timeout behavior to bypass multi-device port authentication and *permit* user access to the network, enter commands similar to the following:

```
FastIron(config)#interface ethernet 3/1
FastIron(config-if-e100-3/1)#dot1x re-auth-timeout-success 60
```

Syntax: [no] dot1x re-auth-timeout- success <seconds>

The <seconds> parameter specifies the number of seconds the device will wait to re-authenticate a user after a timeout. The minimum value is 10 seconds. The maximum value is 2^{16}-1 (maximum unsigned 16-bit value).

Deny User Access to the Network after a RADIUS Timeout

To set the RADIUS timeout behavior to bypass 802.1X authentication and *block* user access to the network, enter commands such as the following:

```
FastIron(config)#interface ethernet 3/1
FastIron(config-if-e100-3/1)#dot1x auth-timeout-action failure
```

Syntax: [no] dot1x auth-timeout-action failure

Once the *failure* timeout action is enabled, use the **no** form of the command to reset the RADIUS timeout behavior to ***retry***.

NOTE: If **restrict-vlan** is configured along with **auth-timeout-action failure**, the user will be placed into a VLAN with restricted or limited access. See “Allow User Access to a Restricted VLAN after a RADIUS Timeout” .

Allow User Access to a Restricted VLAN after a RADIUS Timeout

To set the RADIUS timeout behavior to bypass 802.1X authentication and place the user in a VLAN with restricted or limited access, enter commands such as the following:

```
FastIron(config)#interface ethernet 3/1
FastIron(config-if-e100-3/1)#dot1x auth-fail-action restrict-vlan 100
FastIron(config-if-e100-3/1)#dot1x auth-timeout-action failure
```

Syntax: [no] dot1x auth-fail-action restrict-vlan [<vlan-id>]

Syntax: [no] dot1x auth-timeout-action failure

Configuring Dynamic VLAN Assignment for 802.1X Ports

When a client successfully completes the EAP authentication process, the Authentication Server (the RADIUS server) sends the Authenticator (the Foundry device) a RADIUS Access-Accept message that grants the client access to the network. The RADIUS Access-Accept message contains attributes set for the user in the user's access profile on the RADIUS server.

If one of the attributes in the Access-Accept message specifies a VLAN identifier, and if this VLAN is available on the Foundry device, the client’s port is moved from its default VLAN to this specified VLAN.

NOTE: This feature is supported on port-based VLANs only. This feature cannot be used to place an 802.1X-enabled port into a Layer 3 protocol VLAN.

Automatic Removal of Dynamic VLAN Assignments for 802.1X Ports

Platform Support:

- FGS and FLS devices running software release 04.1.00 and later

For increased security, this feature removes any association between a port and a dynamically-assigned VLAN when all 802.1x sessions for that VLAN have expired on the port.

NOTE: When a **show run** command is issued during a session, the dynamically-assigned VLAN is not displayed.

Enable 802.1X VLAN ID support by adding the following attributes to a user’s profile on the RADIUS server:

Attribute Name	Type	Value
Tunnel-Type	064	13 (decimal) – VLAN

Attribute Name	Type	Value
Tunnel-Medium-Type	065	6 (decimal) – 802
Tunnel-Private-Group-ID	081	<vlan-name> (string) – either the name or the number of a VLAN configured on the Foundry device.

The device reads the attributes as follows:

- If the Tunnel-Type or the Tunnel-Medium-Type attributes in the Access-Accept message do not have the values specified above, the Foundry device ignores the three Attribute-Value pairs. The client becomes authorized, but the client's port is not dynamically placed in a VLAN.
- If the Tunnel-Type or the Tunnel-Medium-Type attributes in the Access-Accept message do have the values specified above, but there is no value specified for the Tunnel-Private-Group-ID attribute, the client will not become authorized.
- When the Foundry device receives the value specified for the Tunnel-Private-Group-ID attribute, it checks whether the <vlan-name> string matches the name of a VLAN configured on the device. If there is a VLAN on the device whose name matches the <vlan-name> string, then the client's port is placed in the VLAN whose ID corresponds to the VLAN name.
- If the <vlan-name> string does not match the name of a VLAN, the Foundry device checks whether the string, when converted to a number, matches the ID of a VLAN configured on the device. If it does, then the client's port is placed in the VLAN with that ID.
- If the <vlan-name> string does not match either the name or the ID of a VLAN configured on the device, then the client will not become authorized.

The **show interface** command displays the VLAN to which an 802.1X-enabled port has been dynamically assigned, as well as the port from which it was moved (that is, the port's default VLAN). See "Displaying Dynamically Assigned VLAN Information" on page 42-29 for sample output indicating the port's dynamically assigned VLAN.

Dynamic Multiple VLAN Assignment for 802.1X Ports

When you add attributes to a user's profile on the RADIUS server, the <vlan-name> value for the Tunnel-Private-Group-ID attribute can specify the name or number of one or more VLANs configured on the Foundry device.

For example, to specify one VLAN, configure the following for the <vlan-name> value in the Tunnel-Private-Group-ID attribute on the RADIUS server:

```
"10" or "marketing"
```

In this example, the port on which the Client is authenticated is assigned to VLAN 10 or the VLAN named "marketing". The VLAN to which the port is assigned must have previously been configured on the Foundry device.

To specify an untagged VLAN:

```
"U:10" or "U:marketing"
```

When the RADIUS server specifies an untagged VLAN ID, the port's default VLAN ID (or ***PVID***) is changed from the system DEFAULT-VLAN (VLAN 1) to the specified VLAN ID. The port transmits only untagged traffic on its PVID. In this example, the port's PVID is changed from VLAN 1 (the DEFAULT-VLAN) to VLAN 10 or the VLAN named "marketing".

The PVID for a port can be changed only once through RADIUS authentication. For example, if RADIUS authentication for a Client causes a port's PVID to be changed from 1 to 10, and then RADIUS authentication for another Client on the same port specifies that the port's PVID be moved to 20, then the second PVID assignment from the RADIUS server is ignored.

If the link goes down, or the dot1x-mac-session for the Client that caused the initial PVID assignment ages out, then the port reverts back to its original (non-RADIUS-specified) PVID, and subsequent RADIUS authentication can change the PVID assignment for the port.

If a port’s PVID is assigned through the multi-device port authentication feature, and 802.1X authentication subsequently specifies a different PVID, then the PVID specified through 802.1X authentication overrides the PVID specified through multi-device port authentication.

To specify tagged VLANs:

"T:12;T:20" or "T:12;T:marketing"

In this example, the port is added to VLANs 12 and 20 or VLANs 12 and the VLAN named "marketing". When a tagged packet is authenticated, and a list of VLANs is specified on the RADIUS server for the MAC address, then the packet’s tag must match one of the VLANs in the list in order for the Client to be successfully authenticated. If authentication is successful, then the port is added to all of the VLANs specified in the list.

Unlike with a RADIUS-specified untagged VLAN, if the dot1x-mac-session for the Client ages out, the port’s membership in RADIUS-specified tagged VLANs is not changed. In addition, if multi-device port authentication specifies a different list of tagged VLANs, then the port is added to the specified list of VLANs. Membership in the VLANs specified through 802.1X authentication is not changed.

To specify an untagged VLAN and multiple tagged VLANs:

"U:10;T:12;T:marketing"

When the RADIUS server returns a value specifying both untagged and tagged VLAN IDs, the port becomes a dual-mode port, accepting and transmitting both tagged traffic and untagged traffic at the same time. A dual-mode port transmits only untagged traffic on its default VLAN (PVID) and only tagged traffic on all other VLANs.

In this example, the port’s VLAN configuration is changed so that it transmits untagged traffic on VLAN 10, and transmits tagged traffic on VLAN 12 and the VLAN named "marketing".

For a configuration example, see “802.1X Authentication with Dynamic VLAN Assignment” on page 42-37.

Saving Dynamic VLAN Assignments to the Running-Config File

You can configure the Foundry device to save the RADIUS-specified VLAN assignments to the device's running-config file. Enter commands such as the following:

```
FastIron(config)#dot1x-enable
FastIron(config-dot1x)#save-dynamicvlan-to-config
```

Syntax: save-dynamicvlan-to-config

By default, the dynamic VLAN assignments are not saved to the running-config file. Entering the **show running-config** command does not display dynamic VLAN assignments, although they can be displayed with the s**how vlan** and **show authenticated-mac-address detail** commands.

NOTE: When this feature is enabled, issuing the command **write mem** will save any dynamic VLAN assignments to the startup configuration file.

Considerations for Dynamic VLAN Assignment in an 802.1X Multiple-Host Configuration

The following considerations apply when a Client in a 802.1X multiple-host configuration is successfully authenticated, and the RADIUS Access-Accept message specifies a VLAN for the port:

- If the port is not already a member of a RADIUS-specified VLAN, and the RADIUS Access-Accept message specifies the name or ID of a valid VLAN on the Foundry device, then the port is placed in that VLAN.
- If the port is already a member of a RADIUS-specified VLAN, and the RADIUS Access-Accept message specifies the name or ID of a different VLAN, then it is considered an authentication failure. The port’s VLAN membership is not changed.
- If the port is already a member of a RADIUS-specified VLAN, and the RADIUS Access-Accept message specifies the name or ID of that same VLAN, then traffic from the Client is forwarded normally.
- If the RADIUS Access-Accept message specifies the name or ID of a VLAN that does not exist on the Foundry device, then it is considered an authentication failure.

- If the port is a tagged or dual-mode port, and the RADIUS Access-Accept message specifies the name or ID of a valid VLAN on the Foundry device, then the port is placed in that VLAN. If the port is already a member of the RADIUS-specified VLAN, no further action is taken. Note that the Client’s dot1x-mac-session is set to “access-is-allowed” for the RADIUS-specified VLAN only. If traffic from the Client’s MAC address is received on any other VLAN, it is dropped.
- If the RADIUS Access-Accept message does not contain any VLAN information, the Client’s dot1x-mac-session is set to “access-is-allowed”. If the port is already in a RADIUS-specified VLAN, it remains in that VLAN.

Using Dynamic VLAN Assignment with the MAC Port Security Feature

MAC port security allows the Foundry device to learn a limited number of “secure” MAC addresses on an interface. The interface forwards only packets with source MAC addresses that match these secure addresses. If the interface receives a packet with a source MAC address that is different from any of the secure addresses, it is considered a security violation, and subsequent packets from the violating MAC address can be dropped, or the port can be disabled entirely.

If a port is disabled due to a MAC port security violation, 802.1X clients attempting to connect over the port cannot be authorized. In addition, 802.1X clients connecting from non-secure MAC addresses cannot be authorized.

To use 802.1X dynamic VLAN assignment with the MAC port security feature on an interface, you must set the number of secure MAC addresses to two or more. For example:

```
FastIron(config)#int e 3/2
FastIron(config-if-e1000-3/2)#port security
FastIron(config-port-security-e1000-3/2)#maximum 2
FastIron(config-port-security-e1000-3/2)#exit
```

See the “Using the MAC Port Security Feature” on page 43-1 for more information.

Dynamically Applying IP ACLs and MAC Filters to 802.1X Ports

Platform Support:

- FGS and FLS devices running software release 04.1.00 and later

Foundry’s 802.1X implementation supports dynamically applying an IP ACL or MAC address filter to a port, based on information received from an Authentication Server.

When a client/supplicant successfully completes the EAP authentication process, the Authentication Server (the RADIUS server) sends the Authenticator (the Foundry device) a RADIUS Access-Accept message that grants the client access to the network. The RADIUS Access-Accept message contains attributes set for the user in the user's access profile on the RADIUS server.

If the Access-Accept message contains Filter-ID (type 11) and/or Vendor-Specific (type 26) attributes, the Foundry device can use information in these attributes to apply an IP ACL or MAC address filter to the authenticated port. This IP ACL or MAC address filter applies to the port for as long as the client is connected to the network. When the client disconnects from the network, the IP ACL or MAC address filter is no longer applied to the port. If an IP ACL or MAC address filter had been applied to the port prior to 802.1X authentication, it is then re-applied to the port.

The Foundry device uses information in the Filter ID and Vendor-Specific attributes as follows:

- The Filter-ID attribute can specify the number of an existing IP ACL or MAC address filter configured on the Foundry device. In this case, the IP ACL or MAC address filter with the specified number is applied to the port.
- The Vendor-Specific attribute can specify actual syntax for a Foundry IP ACL or MAC address filter, which is then applied to the authenticated port. Configuring a Vendor-Specific attribute in this way allows you to create IP ACLs and MAC filters that apply to individual users; that is, ***per-user*** IP ACLs or MAC address filters.

Configuration Considerations

The following restrictions apply to dynamic IP ACLs or MAC address filters:

- FastIron devices support inbound dynamic IP ACLs only. They do not support outbound dynamic ACLs.

- FastIron devices support inbound Vendor-Specific attributes only. They do not support outbound Vendor-Specific attributes.
- A maximum of one IP ACL can be configured in the inbound direction on an interface.
- MAC address filters cannot be configured in the outbound direction on an interface.
- FastIron devices do not support concurrent operation of MAC address filters and IP ACLS.

Disabling and Enabling Strict Security Mode for Dynamic Filter Assignment

By default, 802.1X dynamic filter assignment operates in ***strict security mode***. When strict security mode is enabled, 802.1X authentication for a port fails if the Filter-ID attribute contains invalid information, or if insufficient system resources are available to implement the per-user IP ACLs or MAC address filters specified in the Vendor-Specific attribute.

When strict security mode is enabled:

- If the Filter-ID attribute in the Access-Accept message contains a value that does not refer to an existing filter (that is, a MAC address filter or IP ACL configured on the device), then the port will not be authenticated, regardless of any other information in the message (for example, if the Tunnel-Private-Group-ID attribute specifies a VLAN on which to assign the port).
- If the Vendor-Specific attribute specifies the syntax for a filter, but there are insufficient system resources to implement the filter, then the port will not be authenticated.
- If the device does not have the system resources available to dynamically apply a filter to a port, then the port will not be authenticated.

NOTE: If the Access-Accept message contains values for both the Filter-ID and Vendor-Specific attributes, then the value in the Vendor-Specific attribute (the per-user filter) takes precedence.

Also, if authentication for a port fails because the Filter-ID attribute referred to a non-existent filter, or there were insufficient system resources to implement the filter, then a Syslog message is generated.

When strict security mode is disabled:

- If the Filter-ID attribute in the Access-Accept message contains a value that does not refer to an existing filter (that is, a MAC address filter or IP ACL configured on the device), then the port is still authenticated, but no filter is dynamically applied to it.
- If the Vendor-Specific attribute specifies the syntax for a filter, but there are insufficient system resources to implement the filter, then the port is still authenticated, but the filter specified in the Vendor-Specific attribute is not applied to the port.

By default, strict security mode is enabled for all 802.1X-enabled interfaces, but you can manually disable or enable it, either globally or for specific interfaces.

To disable strict security mode globally, enter the following commands:

```
FastIron(config)#dot1x-enable
FastIron(config-dot1x)#no global-filter-strict-security
```

After you globally disable strict security mode, you can re-enable it by entering the following command:

```
FastIron(config-dot1x)#global-filter-strict-security
```

Syntax: [no] global-filter-strict-security

To disable strict security mode for a specific interface, enter commands such as the following:

```
FastIron(config)#interface e 1
FastIron(config-if-e1000-1)#dot1x disable-filter-strict-security
```

To re-enable strict security mode for an interface, enter the following command:

```
FastIron(config-if-e1000-1)#no dot1x disable-filter-strict-security
```

Syntax: [no] dot1x disable-filter-strict-security

The output of the **show dot1x** and **show dot1x config** commands has been enhanced to indicate whether strict security mode is enabled or disabled globally and on an interface. See “Displaying the Status of Strict Security Mode” on page 42-30.

Dynamically Applying Existing ACLs or MAC Address Filters

When a port is authenticated using 802.1X security, an IP ACL or MAC address filter that exists in the running-config on the Foundry device can be dynamically applied to the port. To do this, you configure the Filter-ID (type 11) attribute on the RADIUS server. The Filter-ID attribute specifies the name or number of the Foundry IP ACL or MAC address filter.

The following is the syntax for configuring the Filter-ID attribute to refer to a Foundry IP ACL or MAC address filter:

Value	Description
ip.<number>.in	Applies the specified numbered ACL to the 802.1X authenticated port in the inbound direction.
ip.<name>.in	Applies the specified named ACL to the 802.1X authenticated port in the inbound direction.
mac.<number>.in	Applies the specified numbered MAC address filter to the 802.1X authenticated port in the inbound direction.

The following table lists examples of values you can assign to the Filter-ID attribute on the RADIUS server to refer to IP ACLs and MAC address filters configured on a Foundry device.

Possible Values for the Filter ID attribute on the RADIUS server	ACL or MAC address filter configured on the Foundry device
ip.2.in	access-list 2 permit host 36.48.0.3 access-list 2 permit 36.0.0.0 0.255.255.255
ip.102.in	access-list 102 permit ip 36.0.0.0 0.255.255.255 any
ip.fdry_filter.in	ip access-list standard fdry_filter permit host 36.48.0.3
mac.2.in	mac filter 2 permit 3333.3333.3333 ffff.ffff.ffff any etype eq 0800
mac.2.in mac.3.in	mac filter 2 permit 3333.3333.3333 ffff.ffff.ffff any etype eq 0800 mac filter 3 permit 2222.2222.2222 ffff.ffff.ffff any etype eq 0800

Notes

- The <name> in the Filter ID attribute is case-sensitive.
- You can specify only numbered MAC address filters in the Filter ID attribute. Named MAC address filters are not supported.
- Dynamic ACL filters are supported only for the inbound direction. Dynamic outbound ACL filters are not supported.
- MAC address filters are supported only for the inbound direction. Outbound MAC address filters are not supported.

- Dynamically assigned IP ACLs and MAC address filters are subject to the same configuration restrictions as non-dynamically assigned IP ACLs and MAC address filters.

Configuring Per-User IP ACLs or MAC Address Filters

Per-user IP ACLs and MAC address filters make use of the Vendor-Specific (type 26) attribute to dynamically apply filters to ports. Defined in the Vendor-Specific attribute are Foundry ACL or MAC address filter statements. When the RADIUS server returns the Access-Accept message granting a client access to the network, the Foundry device reads the statements in the Vendor-Specific attribute and applies these IP ACLs or MAC address filters to the client's port. When the client disconnects from the network, the dynamically applied filters are no longer applied to the port. If any filters had been applied to the port previous to the client connecting, then those filters are reapplied to the port.

The following table shows the syntax for configuring the Foundry Vendor-Specific attributes with ACL or MAC address filter statements:

Value	Description
ipacl.e.in=<extended-acl-entries>	Applies the specified extended ACL entries to the 802.1X authenticated port in the inbound direction.
macfilter.in=<mac-filter-entries>	Applies the specified MAC address filter entries to the 802.1X authenticated port in the inbound direction.

The following table shows examples of IP ACLs and MAC address filters configured in the Foundry Vendor-Specific attribute on a RADIUS server. These IP ACLs and MAC address filters follow the same syntax as other Foundry ACLs and MAC address filters. See the related chapters in this book for information on syntax.

ACL or MAC address filter	Vendor-Specific attribute on RADIUS server
MAC address filter with one entry	macfilter.in= deny any any

MAC address filter with two entries	macfilter.in= permit 0000.0000.3333 ffff.ffff.0000 any,
	macfilter.in= permit 0000.0000.4444 ffff.ffff.0000 any

The RADIUS server allows one instance of the Vendor-Specific attribute to be sent in an Access-Accept message. However, the Vendor-Specific attribute can specify multiple IP ACLs or MAC address filters. You can use commas, semicolons, or carriage returns to separate the filters (for example: ipacl.e.in= permit ip any any,ipacl.e.in = deny ip any any).

Enabling 802.1X Port Security

By default, 802.1X port security is disabled on Foundry devices. To enable the feature on the device and enter the dot1x configuration level, enter the following command:

```
FastIron(config)#dot1x-enable
FastIron(config-dot1x)#
```

Syntax: [no] dot1x-enable

At the dot1x configuration level, you can enable 802.1X port security on all interfaces at once, on individual interfaces, or on a range of interfaces.

For example, to enable 802.1X port security on all interfaces on the device, enter the following command:

```
FastIron(config-dot1x)#enable all
```

Syntax: [no] enable all

To enable 802.1X port security on interface 3/11, enter the following command:

```
FastIron(config-dot1x)#enable ethernet 3/11
```

Syntax: [no] enable ethernet [<slotnum>/]<portnum>

The <slotnum> parameter is required on chassis devices.

The <portnum> parameter is a valid port number.

To enable 802.1X port security on interfaces 3/11 through 3/16, enter the following command:

```
FastIron(config-dot1x)#enable ethernet 3/11 to 3/16
```

Syntax: [no] enable ethernet [<slotnum>/]<portnum> to <portnum>

The <slotnum> parameter is required on chassis devices.

The <portnum> parameter is a valid port number.

Setting the Port Control

To activate authentication on an 802.1X-enabled interface, you specify the kind of ***port control*** to be used on the interface. An interface used with 802.1X port security has two virtual access points: a controlled port and an uncontrolled port.

- The controlled port can be either the authorized or unauthorized state. In the authorized state, it allows normal traffic to pass between the Client and the Authenticator. In the unauthorized state, no traffic is allowed to pass.
- The uncontrolled port allows only EAPOL traffic between the Client and the Authentication Server.

See Figure 42.3 on page 42-3 for an illustration of this concept.

By default, all controlled ports on the device are in the authorized state, allowing all traffic. When you activate authentication on an 802.1X-enabled interface, its controlled port is placed in the unauthorized state. When a Client connected to the interface is successfully authenticated, the controlled port is then placed in the authorized state. The controlled port remains in the authorized state until the Client logs off.

To activate authentication on an 802.1X-enabled interface, you configure the interface to place its controlled port in the authorized state when a Client is authenticated by an Authentication Server. To do this, enter commands such as the following.

```
FastIron(config)#interface e 3/1
FastIron(config-if-3/1)#dot1x port-control auto
```

Syntax: [no] dot1x port-control [force-authorized | force-unauthorized | auto]

When an interface's control type is set to **auto**, the controlled port is initially set to unauthorized, but is changed to authorized when the connecting Client is successfully authenticated by an Authentication Server.

The port control type can be one of the following:

force-authorized – The controlled port is placed unconditionally in the authorized state, allowing all traffic. This is the default state for ports on the Foundry device.

force-unauthorized – The controlled port is placed unconditionally in the unauthorized state.

auto – The controlled port is unauthorized until authentication takes place between the Client and Authentication Server. Once the Client passes authentication, the port becomes authorized. This activates authentication on an 802.1X-enabled interface.

NOTE: You cannot enable 802.1X port security on ports that have any of the following features enabled:

- Link aggregation

- Metro Ring Protocol (MRP)
- Tagged port
- Mirror port
- Trunk port

Configuring Periodic Re-Authentication

You can configure the device to periodically re-authenticate Clients connected to 802.1X-enabled interfaces. When you enable periodic re-authentication, the device re-authenticates Clients every 3,600 seconds by default. You can optionally specify a different re-authentication interval of between 1 – 4294967295 seconds.

To configure periodic re-authentication using the default interval of 3,600 seconds, enter the following command:

```
FastIron(config-dot1x)#re-authentication
```

Syntax: [no] re-authentication

To configure periodic re-authentication with an interval of 2,000 seconds, enter the following commands:

```
FastIron(config-dot1x)#re-authentication
FastIron(config-dot1x)#timeout re-authperiod 2000
```

Syntax: [no] timeout re-authperiod <seconds>

The re-authentication interval is a global setting, applicable to all 802.1X-enabled interfaces. To re-authenticate Clients connected to a specific port manually, use the **dot1x re-authenticate** command. See “Re-Authenticating a Port Manually” , below.

Re-Authenticating a Port Manually

When periodic re-authentication is enabled, by default the Foundry device re-authenticates Clients connected to an 802.1X-enabled interface every 3,600 seconds (or the time specified by the **dot1x timeout re-authperiod** command). You can also manually re-authenticate Clients connected to a specific port.

For example, to re-authenticate Clients connected to interface 3/1, enter the following command:

```
FastIron#dot1x re-authenticate e 3/1
```

Syntax: dot1x re-authenticate ethernet [<slotnum>/]<portnum>

The <slotnum> parameter is required on chassis devices.

The <portnum> parameter is a valid port number.

Setting the Quiet Period

If the Foundry device is unable to authenticate the Client, the Foundry device waits a specified amount of time before trying again. The amount of time the Foundry device waits is specified with the **quiet-period** parameter. The **quiet-period** parameter can be from 0 – 4294967295 seconds. The default is 60 seconds.

For example, to set the quiet period to 30 seconds, enter the following command:

```
FastIron(config-dot1x)#timeout quiet-period 30
```

Syntax: [no] timeout quiet-period <seconds>

Specifying the Wait Interval and Number of EAP-Request/Identity Frame Retransmissions from the Foundry Device

When the Foundry device sends an EAP-request/identity frame to a Client, it expects to receive an EAP-response/identity frame from the Client. By default, if the Foundry device does not receive an EAP-response/identity frame from a Client, the device waits 30 seconds, then retransmits the EAP-request/identity frame. Also by default, the

Foundry device retransmits the EAP-request/identity frame a maximum of two times. You can optionally configure the amount of time the device will wait before retransmitting an EAP-request/identity frame, and the number of times the EAP-request/identity frame will be transmitted. This section provides the command syntax for these features.

Setting the Wait Interval for EAP Frame Retransmissions

By default, if the Foundry device does not receive an EAP-response/identity frame from a Client, the device waits 30 seconds, then retransmits the EAP-request/identity frame. You can optionally change the amount of time the Foundry device waits before retransmitting the EAP-request/identity frame to the Client.

For example, to cause the Foundry device to wait 60 seconds before retransmitting an EAP-request/identity frame to a Client, enter the following command:

```
FastIron(config-dot1x)#timeout tx-period 60
```

If the Client does not send back an EAP-response/identity frame within 60 seconds, the device will transmit another EAP-request/identity frame.

Syntax: [no] timeout tx-period <seconds>

where <seconds> is a value from 0 – 4294967295. The default is 30 seconds.

Setting the Maximum Number of EAP Frame Retransmissions

The Foundry device retransmits the EAP-request/identity frame a maximum of two times. If no EAP-response/identity frame is received from the Client after two EAP-request/identity frame retransmissions (or the amount of time specified with the **auth-max** command), the device restarts the authentication process with the Client.

You can optionally change the number of times the Foundry device should retransmit the EAP-request/identity frame. You can specify between 1 – 10 frame retransmissions. For example, to configure the device to retransmit an EAP-request/identity frame to a Client a maximum of three times, enter the following command:

FESX and FWSX devices running releases 02.2.00 and later and the FSX devices running 02.3.01 and later, use the **auth-max** command in lieu of the **maxreq** command, such as the following:

```
FastIron(config-dot1x)#auth-max 3
```

Syntax: auth-max <value>

<value> is a number from 1 – 10. The default is 2.

Specifying the Wait Interval and Number of EAP-Request/Identity Frame Retransmissions from the RADIUS Server

Acting as an intermediary between the RADIUS Authentication Server and the Client, the Foundry device receives RADIUS messages from the RADIUS server, encapsulates them as EAPOL frames, and sends them to the Client. By default, when the Foundry device relays an EAP-Request frame from the RADIUS server to the Client, it expects to receive a response from the Client within 30 seconds. If the Client does not respond within the allotted time, the device retransmits the EAP-Request frame to the Client. Also by default, the Foundry device retransmits the EAP-request frame twice. If no EAP-response frame is received from the Client after two EAP-request frame retransmissions, the device restarts the authentication process with the Client.

You can optionally configure the amount of time the device will wait before retransmitting an EAP-request/identity frame, and the number of times the EAP-request/identity frame will be transmitted. This section provides the command syntax for these features.

Setting the Wait Interval for EAP Frame Retransmissions

By default, when the Foundry device relays an EAP-Request frame from the RADIUS server to the Client, it expects to receive a response from the Client within 30 seconds. You can optionally specify the wait interval using the **supptimeout** command.

For example, to configure the device to retransmit an EAP-Request frame if the Client does not respond within 45 seconds, enter the following command:

```
FastIron(config-dot1x)#supptimeout 45
```

Syntax: supptimeout <seconds>

<seconds> is a number from 0 – 4294967295 seconds. The default is 30 seconds.

Setting the Maximum Number of EAP Frame Retransmissions

You can optionally specify the number of times the Foundry device will retransmit the EAP-request frame. You can specify between 1 – 10 frame retransmissions. For example, to configure the device to retransmit an EAP-request frame to a Client a maximum of three times, enter the following command:

```
FastIron(config-dot1x)#max-req 3
```

Syntax: max-req <value>

<value> is a number from 1 – 10. The default is 2.

Specifying a Timeout for Retransmission of Messages to the Authentication Server

When performing authentication, the Foundry device receives EAPOL frames from the Client and passes the messages on to the RADIUS server. The device expects a response from the RADIUS server within 30 seconds. If the RADIUS server does not send a response within 30 seconds, the Foundry device retransmits the message to the RADIUS server. The time constraint for retransmission of messages to the Authentication Server can be between 0 – 4294967295 seconds.

For example, to configure the device to retransmit a message if the Authentication Server does not respond within 45 seconds, enter the following command:

```
FastIron(config-dot1x)#servertimeout 45
```

Syntax: servertimeout <seconds>

Initializing 802.1X on a Port

To initialize 802.1X port security on a port, enter a command such as the following:

```
FastIron#dot1x initialize e 3/1
```

Syntax: dot1x initialize ethernet [<slotnum>/]<portnum>

The <slotnum> parameter is required on chassis devices.

The <portnum> parameter is a valid port number.

Allowing Access to Multiple Hosts

Foundry devices support 802.1X authentication for ports with more than one host connected to them. If there are multiple hosts connected to a single 802.1X-enabled port, the Foundry device authenticates each of them individually. See “Configuring 802.1X Multiple-Host Authentication” .

Configuring 802.1X Multiple-Host Authentication

When multiple hosts are connected to the same 802.1X-enabled port, the functionality described in “How 802.1X Multiple-Host Authentication Works” on page 42-6 is enabled by default. You can optionally do the following:

- Specify the authentication-failure action
- Specify the number of authentication attempts the device makes before dropping packets
- Disabling aging for dot1x-mac-sessions
- Configure aging time for blocked Clients
- Clear the dot1x-mac-session for a MAC address

Specifying the Authentication-Failure Action

In an 802.1X multiple-host configuration, if RADIUS authentication for a Client is unsuccessful, traffic from that Client is either dropped in hardware (the default), or the Client’s port is placed in a “restricted” VLAN. You can specify which of these two authentication-failure actions is to be used. If the authentication-failure action is to place the port in a restricted VLAN, you can specify the ID of the restricted VLAN.

To specify that the authentication-failure action is to place the Client's port in a restricted VLAN, enter the following command:

```
FastIron(config)#dot1x-enable
FastIron(config-dot1x)#auth-fail-action restricted-vlan
```

Syntax: [no] auth-fail-action restricted-vlan

To specify the ID of the restricted VLAN as VLAN 300, enter the following command:

```
FastIron(config-dot1x)#auth-fail-vlanid 300
```

Syntax: [no] auth-fail-vlanid <vlan-id>

Specifying the Number of Authentication Attempts the Device Makes Before Dropping Packets

When the authentication-failure action is to drop traffic from the Client, and the initial authentication attempt made by the device to authenticate the Client is unsuccessful, the Foundry device immediately retries to authenticate the Client. After three unsuccessful authentication attempts, the Client's dot1x-mac-session is set to “access-denied”, causing traffic from the Client to be dropped in hardware.

You can optionally configure the number of authentication attempts the device makes before dropping traffic from the Client. To do so, enter a command such as the following:

```
FastIron(config-dot1x)#auth-fail-max-attempts 2
```

Syntax: [no] auth-fail-max-attempts <attempts>

By default, the device makes 3 attempts to authenticate a Client before dropping packets from the Client. You can specify between 1 – 10 authentication attempts.

Disabling Aging for dot1x-mac-sessions

The dot1x-mac-sessions for Clients authenticated or denied by a RADIUS server are aged out if no traffic is received from the Client's MAC address for a certain period of time. After a Client's dot1x-mac-session is aged out, the Client must be re-authenticated.

- ***Permitted*** dot1x-mac-sessions, which are the dot1x-mac-sessions for authenticated Clients, as well as for non-authenticated Clients whose ports have been placed in the restricted VLAN, are aged out if no traffic is received from the Client's MAC address over the Foundry device's normal MAC aging interval.
- ***Denied*** dot1x-mac-sessions, which are the dot1x-mac-sessions for non-authenticated Clients that are blocked by the Foundry device are aged out over a configurable software aging period. (See the next section for more information on configuring the software aging period).

You can optionally disable aging of the permitted and/or denied dot1x-mac-sessions on the Foundry device.

To disable aging of the permitted dot1x-mac-sessions, enter the following command:

```
FastIron(config-dot1x)#mac-session-aging no-aging permitted-mac-only
```

Syntax: [no] mac-session-aging no-aging permitted-mac-only

To disable aging of the denied dot1x-mac-sessions, enter the following command:

```
FastIron(config-dot1x)#mac-session-aging no-aging denied-mac-only
```

Syntax: [no] mac-session-aging no-aging denied-mac-only

NOTE: This command enables aging of permitted sessions.

As a shortcut, use the command **[no] mac-session-aging** to enable or disable aging for permitted and denied sessions.

Specifying the Aging Time for Blocked Clients

When the Foundry device is configured to drop traffic from non-authenticated Clients, traffic from the blocked Clients is dropped in hardware, without being sent to the CPU. A Layer 2 CAM entry is created that drops traffic from the blocked Client’s MAC address in hardware. If no traffic is received from the blocked Client’s MAC address for a certain amount of time, this Layer 2 CAM entry is aged out. If traffic is subsequently received from the Client’s MAC address, then an attempt can be made to authenticate the Client again.

Aging of the Layer 2 CAM entry for a blocked Client’s MAC address occurs in two phases, known as ***hardware aging*** and ***software aging***. The hardware aging period is fixed at 70 seconds and is non-configurable. The software aging time is configurable through the CLI.

Once the Foundry device stops receiving traffic from a blocked Client’s MAC address, the hardware aging begins and lasts for a fixed period of time. After the hardware aging period ends, the software aging period begins. The software aging period lasts for a configurable amount of time (by default 120 seconds). After the software aging period ends, the blocked Client’s MAC address ages out, and can be authenticated again if the Foundry device receives traffic from the Client’s MAC address.

Change the length of the software aging period for a blocked Client’s MAC address by entering a command such as the following:

```
FastIron(config)#mac-session-aging max-age 180
```

Syntax: [no] mac-session-aging max-age <seconds>

You can specify from 1 – 65535 seconds. The default is 120 seconds.

Clearing a dot1x-mac-session for a MAC Address

You can clear the dot1x-mac-session for a specified MAC address, so that the Client with that MAC address can be re-authenticated by the RADIUS server. For example:

```
FastIron#clear dot1x mac-session 00e0.1234.abd4
```

Syntax: clear dot1x mac-session <mac-address>

Defining MAC Filters for EAP Frames

You can create MAC address filters to permit or deny EAP frames. To do this, you specify the Foundry device’s 802.1X group MAC address as the destination address in a MAC filter, then apply the filter to an interface.

MAC Filters for EAPS on most devices

For example, the following command creates a MAC filter that denies frames with the destination MAC address of 0180.c200.0003, which is the Foundry device’s 802.1X group MAC address:

```
FastIron(config)#mac filter 1 deny any 0180.c200.0003 ffff.ffff.ffff
```

The following commands apply this filter to interface e 3/1:

```
FastIron(config)#interface e 3/11
FastIron(config-if-3/1)#mac filter-group 1
```

See “Defining MAC Address Filters” on page 44-6 for more information.

Configuring Guest VLAN Access for Non-EAP-Capable Clients

You can configure the Foundry device to grant "guest" VLAN access to clients that do not support Extensible Authentication Protocol (EAP). The guest VLAN (also called restricted VLAN) limits access to the network or applications, instead of blocking access to these services altogether.

When the Foundry device receives the first packet (non-EAP packet) from a client, the device waits for 10 seconds or the amount of time specified with the **timeout restrict-fwd-period** command. If the Foundry device does not receive subsequent packets after the timeout period, the device places the client onto the guest VLAN.

This feature is disabled by default. To enable this feature and change the timeout period, enter commands such as the following:

```
FastIron(config)#dot1x-enable
FastIron(config-dot1x)#restrict-forward-non-dot1x
FastIron(config-dot1x)#timeout restrict-fwd-period 15
```

Syntax: restrict-forward-non-dot1x

Syntax: timeout restrict-fwd-period <num>

The <num> parameter is a value from 0 to 4294967295. The default value is 10.

Displaying 802.1X Information

You can display the following 802.1X-related information:

- The 802.1X configuration on the device and on individual ports
- Statistics about the EAPOL frames passing through the device
- 802.1X-enabled ports dynamically assigned to a VLAN
- User-defined and dynamically applied MAC filters and IP ACLs currently active on the device
- The 802.1X multiple-host configuration

Displaying 802.1X Configuration Information

To display information about the 802.1X configuration on the Foundry device, enter the following command:

```
FastIron#show dot1x
PAE Capability:    Authenticator Only
system-auth-control: Enable
re-authentication: Disable
global-filter-strict-security: Enable
quiet-period:    60 Seconds
tx-period:    30 Seconds
supptimeout:    30 Seconds
servertimeout:     30 Seconds
maxreq:    2
re-authperiod:    3600 Seconds
Protocol Version:    1
```

Syntax: show dot1x

The following table describes the information displayed by the **show dot1x** command.

Table 42.1: Output from the show dot1x command

This Field...	Displays...
PAE Capability	The Port Access Entity (PAE) role for the Foundry device. This is always “Authenticator Only”.
system-auth-control	Whether system authentication control is enabled on the device. The **dot1x-enable** command enables system authentication control on the device.
re-authentication	Whether periodic re-authentication is enabled on the device. See “Configuring Periodic Re-Authentication” on page 42-19. When periodic re-authentication is enabled, the device automatically re-authenticates Clients every 3,600 seconds by default.
global-filter-strict-security:	Whether strict security mode is enabled or disabled globally. See “Disabling and Enabling Strict Security Mode for Dynamic Filter Assignment” on page 42-15.

Table 42.1: Output from the show dot1x command (Continued)

This Field...	Displays...
quiet-period	When the Foundry device is unable to authenticate a Client, the amount of time the Foundry device waits before trying again (default 60 seconds). See “Setting the Quiet Period” on page 42-19 for information on how to change this setting.
tx-period	When a Client does not send back an EAP-response/identity frame, the amount of time the Foundry device waits before retransmitting the EAP-request/identity frame to a Client (default 30 seconds). See “Setting the Wait Interval for EAP Frame Retransmissions” on page 42-20 for information on how to change this setting.
supp-timeout	When a Client does not respond to an EAP-request frame, the amount of time before the Foundry device retransmits the frame. See “Setting the Wait Interval for EAP Frame Retransmissions” on page 42-20 for information on how to change this setting.
server-timeout	When the Authentication Server does not respond to a message sent from the Client, the amount of time before the Foundry device retransmits the message. See “Specifying a Timeout for Retransmission of Messages to the Authentication Server” on page 42-21 for information on how to change this setting.
max-req	The number of times the Foundry device retransmits an EAP-request/identity frame if it does not receive an EAP-response/identity frame from a Client (default 2 times). See “Setting the Maximum Number of EAP Frame Retransmissions” on page 42-20 for information on how to change this setting.
re-authperiod	How often the device automatically re-authenticates Clients when periodic re-authentication is enabled (default 3,600 seconds). See “Configuring Periodic Re-Authentication” on page 42-19 for information on how to change this setting.
Protocol Version	The version of the 802.1X protocol in use on the device.

To display information about the 802.1X configuration on an individual port, enter a command such as the following:

```
FastIron#show dot1x configuration ethernet 1/3
Port-Control                    : control-auto
filter strict security          : Enable
Action on RADIUS timeout        : Treat as a failed authentication
  re-authenticate               : 150 seconds
PVID State                      : Normal (101)
Original PVID                   : 101
PVID mac total                  : 1
PVID mac authorized             : 1
num mac sessions                : 1
num mac authorized              : 1
Number of Auth filter           : 0
```

Syntax: show dot1x config ethernet [<slotnum>/]<portnum>

The <slotnum> parameter is required on chassis devices.

The <portnum> parameter is a valid port number.

The following additional information is displayed in the **show dot1x config** command for an interface:

Table 42.2: Output from the show dot1x config command for an interface

This Field...	Displays...
Authenticator PAE state	The current status of the Authenticator PAE state machine. This can be INITIALIZE, DISCONNECTED, CONNECTING, AUTHENTICATING, AUTHENTICATED, ABORTING, HELD, FORCE_AUTH, or FORCE_UNAUTH. **Note:** When the Authenticator PAE state machine is in the AUTHENTICATING state, if the reAuthenticate, eapStart, eapLogoff, or authTimeout parameters are set to TRUE, it may place the Authenticator PAE state machine indefinitely in the ABORTING state. If this should happen, use the **dot1x initialize** command to initialize 802.1X port security on the port, or unplug the Client or hub connected to the port, then reconnect it.
Backend Authentication state	The current status of the Backend Authentication state machine. This can be REQUEST, RESPONSE, SUCCESS, FAIL, TIMEOUT, IDLE, or INITIALIZE.
AdminControlledDirections	Indicates whether an unauthorized controlled port exerts control over communication in both directions (disabling both reception of incoming frames and transmission of outgoing frames), or just in the incoming direction (disabling only reception of incoming frames). On Foundry devices, this parameter is set to BOTH.

Table 42.2: Output from the show dot1x config command for an interface (Continued)

This Field...	Displays...
OperControlledDirections	The setting for the OperControlledDirections parameter, as defined in the 802.1X standard. According to the 802.1X standard, if the AdminControlledDirections parameter is set to BOTH, the OperControlledDirections parameter is unconditionally set to BOTH. Since the AdminControlledDirections parameter on Foundry devices is always set to BOTH, the OperControlledDirections parameter is also set to BOTH.
AuthControlledPortControl	The port control type configured for the interface. If set to auto, authentication is activated on the 802.1X-enabled interface.
AuthControlledPortStatus	The current status of the interface's controlled port: either authorized or unauthorized.
multiple-hosts	Whether the port is configured to allow multiple Supplicants accessing the interface on the Foundry device through a hub. See "Allowing Access to Multiple Hosts" on page 42-21 for information on how to change this setting.

Displaying 802.1X Statistics

To display 802.1X statistics for an individual port, enter a command such as the following:

```
FastIron#show dot1x statistics e 3/3

Port 3/3 Statistics:
RX EAPOL Start:    0
RX EAPOL Logoff:    0
RX EAPOL Invalid:    0
RX EAPOL Total:    0
RX EAP Resp/Id:    0
RX EAP Resp other than Resp/Id:    0
RX EAP Length Error:    0
Last EAPOL Version:    0
Last EAPOL Source:    0007.9550.0B83
TX EAPOL Total:    217
TX EAP Req/Id:    163
TX EAP Req other than Req/Id:    0
```

Syntax: show dot1x statistics ethernet [<slotnum>/]<portnum>

The <slotnum> parameter is required on chassis devices.

The <portnum> parameter is a valid port number.

The following table describes the information displayed by the **show dot1x statistics** command for an interface.

Table 42.3: Output from the show dot1x statistics command

This Field...	Displays...
RX EAPOL Start	The number of EAPOL-Start frames received on the port.
RX EAPOL Logoff	The number of EAPOL-Logoff frames received on the port.
RX EAPOL Invalid	The number of invalid EAPOL frames received on the port.
RX EAPOL Total	The total number of EAPOL frames received on the port.
RX EAP Resp/Id	The number of EAP-Response/Identity frames received on the port
RX EAP Resp other than Resp/Id	The total number of EAPOL-Response frames received on the port that were not EAP-Response/Identity frames.
RX EAP Length Error	The number of EAPOL frames received on the port that have an invalid packet body length.
Last EAPOL Version	The version number of the last EAPOL frame received on the port.
Last EAPOL Source	The source MAC address in the last EAPOL frame received on the port.
TX EAPOL Total	The total number of EAPOL frames transmitted on the port.
TX EAP Req/Id	The number of EAP-Request/Identity frames transmitted on the port.
TX EAP Req other than Req/Id	The number of EAP-Request frames transmitted on the port that were not EAP-Request/Identity frames.

Clearing 802.1X Statistics

You can clear the 802.1X statistics counters on all interfaces at once, on individual interfaces, or on a range of interfaces.

For example, to clear the 802.1X statistics counters on all interfaces on the device, enter the following command:

```
FastIron#clear dot1x statistics all
```

Syntax: clear dot1x statistics all

To clear the 802.1X statistics counters on interface e 3/11, enter the following command:

```
FastIron#clear dot1x statistics e 3/11
```

Syntax: clear dot1x statistics ethernet [<slotnum>/]<portnum>

The <slotnum> parameter is required on chassis devices.

The <portnum> parameter is a valid port number.

Displaying Dynamically Assigned VLAN Information

The **show interface** command displays the VLAN to which an 802.1X-enabled port has been dynamically assigned, as well as the port from which it was moved (that is, the port's default VLAN).

The following example of the **show interface** command indicates the port's dynamically assigned VLAN. Information about the dynamically assigned VLAN is shown in bold type.

```
FastIron#show interface e 12/2
FastEthernet12/2 is up, line protocol is up
  Hardware is FastEthernet, address is 0204.80a0.4681 (bia 0204.80a0.4681)
  Configured speed auto, actual 100Mbit, configured duplex fdx, actual fdx
  Member of L2 VLAN ID 2 (dot1x-RADIUS assigned), original L2 VLAN ID is 1,
  port is untagged, port state is FORWARDING
  STP configured to ON, priority is level0, flow control enabled
  mirror disabled, monitor disabled
  Not member of any active trunks
  Not member of any configured trunks
  No port name
  MTU 1518 bytes
  300 second input rate: 0 bits/sec, 0 packets/sec, 0.00% utilization
  300 second output rate: 256 bits/sec, 0 packets/sec, 0.00% utilization
  3 packets input, 192 bytes, 0 no buffer
  Received 0 broadcasts, 0 multicasts, 3 unicasts
  0 input errors, 0 CRC, 0 frame, 0 ignored
  0 runts, 0 giants, DMA received 3 packets
  919 packets output, 58816 bytes, 0 underruns
  Transmitted 1 broadcasts, 916 multicasts, 2 unicasts
  0 output errors, 0 collisions, DMA transmitted 919 packets
```

In this example, the 802.1X-enabled port has been moved from VLAN 1 to VLAN 2. When the client disconnects, the port will be moved back to VLAN 1.

The **show run** command also indicates the VLAN to which the port has been dynamically assigned. The output can differ depending on whether GARP VLAN Registration Protocol (GVRP) is enabled on the device:

- **Without GVRP** – When you enter the **show run** command, the output indicates that the port is a member of the VLAN to which it was dynamically assigned through 802.1X. If you then enter the **write memory** command, the VLAN to which the port is currently assigned becomes the port's default VLAN in the device's configuration.
- **With GVRP** – When you enter the **show run** command, if the VLAN name supplied by the RADIUS server corresponds to a VLAN learned through GVRP, then the output indicates that the port is a member of the VLAN to which it was originally assigned (not the VLAN to which it was dynamically assigned).

  If the VLAN name supplied by the RADIUS server corresponds to a statically configured VLAN, the output indicates that the port is a member of the VLAN to which it was dynamically assigned through 802.1X. If you then enter the **write memory** command, the VLAN to which the port is currently assigned becomes the port's default VLAN in the device's configuration.

Displaying Information About Dynamically Applied MAC Filters and IP ACLs

You can display information about currently active user-defined and dynamically applied MAC filters and IP ACLs.

Displaying User-Defined MAC Filters and IP ACLs

To display the user-defined MAC filters active on the device, enter the following command:

```
FastIron#show dot1x mac-address filter

Port 1/3 (User defined MAC Address Filter) :
      mac filter 1 permit any any
```

Syntax: show dot1x mac-address-filter

To display the user-defined IP ACLs active on the device, enter the following command:

```
FastIron#show dot1x ip-acl

Port 1/3 (User defined IP ACLs):

Extended IP access list Port_1/3_E_IN
permit udp any any

Extended IP access list Port_1/3_E_OUT
permit udp any any
```

Syntax: show dot1x ip-acl

Displaying Dynamically Applied MAC Filters and IP ACLs

To display the dynamically applied MAC address filters active on an interface, enter a command such as the following:

```
FastIron#show dot1x mac-address-filter e 1/3

Port 1/3 MAC Address Filter information:
  802.1X Dynamic MAC Address Filter :
     mac filter-group 2
  Port default MAC Address Filter:
     No mac address filter is set
```

Syntax: show dot1x mac-address-filter all | ethernet [<slotnum>/]<portnum>

The **all** keyword displays all dynamically applied MAC address filters active on the device.

The <slotnum> parameter is required on chassis devices.

The <portnum> parameter is a valid port number.

To display the dynamically applied IP ACLs active on an interface, enter a command such as the following:

```
FastIron#show dot1x ip-acl e 1/3

Port 1/3 IP ACL information:
  802.1X dynamic IP ACL (user defined) in:
    ip access-list extended Port_1/3_E_IN in
  Port default IP ACL in:
    No inbound ip access-list is set
  802.1X dynamic IP ACL (user defined) out:
    ip access-list extended Port_1/3_E_OUT out
  Port default IP ACL out:
    No outbound ip access-list is set
```

Syntax: show dot1x ip-acl all | ethernet [<slotnum>/]<portnum>

The **all** keyword displays all dynamically applied IP ACLs active on the device.

The <slotnum> parameter is required on chassis devices.

The <portnum> parameter is a valid port number.

Displaying the Status of Strict Security Mode

Platform Support:

- FESX/FSX/FWSX devices running software release 02.2.00 and later

The output of the **show dot1x** and **show dot1x config** commands indicate whether strict security mode is enabled or disabled globally and on an interface.

Displaying the Status of Strict Security Mode Globally on the Device

To display the status of strict security mode globally on the device, enter the following command:

```
FastIron#show dot1x
PAE Capability:    Authenticator Only
system-auth-control: Enable
re-authentication: Disable
global-filter-strict-security: Enable
quiet-period:    60 Seconds
tx-period:    30 Seconds
supptimeout:    30 Seconds
servertimeout:     30 Seconds
maxreq:    2
re-authperiod:    3600 Seconds
security-hold-time: 60 Seconds
Protocol Version:    1
```

Syntax: show dot1x

Displaying the Status of Strict Security Mode on an Interface

To display the status of strict security mode on an interface, enter a command such as the following:

```
FastIron#show dot1x config e 1/3

Port 1/3 Configuration:
Authenticator PAE state:    AUTHENTICATED
Backend Authentication state:     IDLE
AdminControlledDirections:    BOTH
OperControlledDirections:    BOTH
AuthControlledPortControl:     Auto
AuthControlledPortStatus:    authorized
quiet-period:    60 Seconds
tx-period:    30 Seconds
supptimeout:    30 Seconds
servertimeout:     30 Seconds
maxreq:    2
re-authperiod:    3600 Seconds
security-hold-time: 60 Seconds
re-authentication: Disable
multiple-hosts: Disable
filter-strict-security: Enable
Protocol Version:    1
```

Syntax: show dot1x config ethernet [<slotnum>/]<portnum>

The <slotnum> parameter is required on chassis devices.

The <portnum> parameter is a valid port number.

Displaying 802.1X Multiple-Host Authentication Information

You can display the following information about 802.1X multiple-host authentication:

- Information about the 802.1X multiple-host configuration
- The dot1x-mac-sessions on each port
- The number of users connected on each port in a 802.1X multiple-host configuration

Displaying 802.1X Multiple-Host Configuration Information

The output of the **show dot1x** and **show dot1x config** commands displays information related to 802.1X multiple-host authentication.

The following is an example of the output of the **show dot1x** command. The information related to multiple-host authentication is highlighted in bold.

```
FastIron#show dot1x

Number of Ports enabled          : 2
Re-Authentication                : Enabled
Authentication-fail-action       : Restricted VLAN
Authentication Failure VLAN      : 111
Mac Session Aging                : Disabled for permitted MAC sessions
Mac Session max-age              : 60 seconds
Protocol Version                 : 1
quiet-period                     : 5 Seconds
tx-period                        : 30 Seconds
supptimeout                      : 30 Seconds
servertimeout                    : 30 Seconds
maxreq                           : 2
re-authperiod                    : 3600 Seconds
security-hold-time               : 60 Seconds
re-authentication                : Enable
Flow based multi-user policy        : Disable
```

Syntax: show dot1x

Table 42.4 describes the bold fields in the display.

Table 42.4: Output from the show dot1x command for multiple host authentication

This Field...	Displays...
Authentication-fail-action	The configured authentication-failure action. This can be Restricted VLAN or Block Traffic.
Authentication Failure VLAN	If the authentication-failure action is Restricted VLAN, the ID of the VLAN to which unsuccessfully authenticated Client ports are assigned.
Mac Session Aging	Whether aging for dot1x-mac-sessions has been enabled or disabled for permitted or denied dot1x-mac-sessions.
Mac Session max-age	The configured software aging time for dot1x-mac-sessions.
Flow based multi-user policy	The dynamically assigned IP ACLs and MAC address filters used in the 802.1X multiple-host configuration.

The output of the **show dot1x config** command for an interface displays the configured port control for the interface. Starting in release 02.2.00, this command also displays information related to 802.1X multiple host-authentication.

The following is an example of the output of the **show dot1x config** command for an interface.

```
FastIron#show dot1x config e 3/1

Port-Control                     : control-auto
filter strict security           : Enable
PVID State                       : Restricted (10)
Original PVID                    : 10
PVID mac total                   : 1
PVID mac authorized              : 0
num mac sessions                 : 1
num mac authorized               : 0
```

Syntax: show dot1x config ethernet [<slotnum>/]<portnum>

The <slotnum> parameter is required on chassis devices.

The <portnum> parameter is a valid port number.

The following table lists the fields in the display.

Table 42.5: Output from the show dot1x config command

This Field...	Displays...
Port-Control	The configured port control type for the interface. This can be one of the following: **force-authorized** – The controlled port is placed unconditionally in the authorized state, allowing all traffic. This is the default state for ports on the Foundry device. **force-unauthorized** – The controlled port is placed unconditionally in the unauthorized state. No authentication takes place for any connected 802.1X Clients. **auto** – The authentication status for each 802.1X Client depends on the authentication status returned from the RADIUS server.
filter strict security	Whether strict security mode is enabled or disabled on the interface.
PVID State	The port's default VLAN ID (PVID) and the state of the port's PVID. The PVID state can be one of the following: **Normal** – The port's PVID is not set by a RADIUS server, nor is it the restricted VLAN. **RADIUS** – The port's PVID was dynamically assigned by a RADIUS server. **RESTRICTED** – The port's PVID is the restricted VLAN.
Original PVID	The originally configured (not dynamically assigned) PVID for the port.
PVID mac total	The number of devices transmitting untagged traffic on the port's PVID.
PVID mac authorized	The number of devices transmitting untagged traffic on the port's PVID as a result of dynamic VLAN assignment.
num mac sessions	The number of dot1x-mac-sessions on the port.
num mac authorized	The number of authorized dot1x-mac-sessions on the port.

Displaying Information About the Dot1x MAC Sessions on Each Port

The **show dot1x mac-session** command displays information about the dot1x-mac-sessions on each port on the device. Starting in software release 02.2.00, the output also shows the authenticator PAE state. For example:

```
FastIron#show dot1x mac-session

Port  MAC/(username)                  Vlan Auth    ACL    Age    PAE
                                           State                 State
------------------------------------------------------------------------------
1     0010.a498.24f7 :User            10   permit  none   S20    AUTHENTICATED
```

Syntax: show dot1x mac-session

Table 42.6 lists the new fields in the display.

Table 42.6: Output from the show dot1x mac-session command

This Field...	Displays...
Port	The port on which the dot1x-mac-session exists.
MAC/ (username)	The MAC address of the Client and the username used for RADIUS authentication.
Vlan	The VLAN to which the port is currently assigned.
Auth-State	The authentication state of the dot1x-mac-session. This can be one of the following: permit – The Client has been successfully authenticated, and traffic from the Client is being forwarded normally. blocked – Authentication failed for the Client, and traffic from the Client is being dropped in hardware. restricted – Authentication failed for the Client, but traffic from the Client is allowed in the restricted VLAN only. init - The Client is in is in the process of 802.1X authentication, or has not started the authentication process.
Age	The software age of the dot1x-mac-session.
PAE State	The current status of the Authenticator PAE state machine. This can be INITIALIZE, DISCONNECTED, CONNECTING, AUTHENTICATING, AUTHENTICATED, ABORTING, HELD, FORCE_AUTH, or FORCE_UNAUTH. **Note:** When the Authenticator PAE state machine is in the AUTHENTICATING state, if the reAuthenticate, eapStart, eapLogoff, or authTimeout parameters are set to TRUE, it may place the Authenticator PAE state machine indefinitely in the ABORTING state. If this should happen, use the dot1x initialize command to initialize 802.1X port security on the port, or unplug the Client or hub connected to the port, then reconnect it.

Displaying Information About the Ports in an 802.1X Multiple-Host Configuration

To display information about the ports in an 802.1X multiple-host configuration, enter the following command:

```
FastIron#show dot1x mac-session brief

Port            Number of  Number of       Dynamic Dynamic
                 users    Authorized users  VLAN   Filters
----------------------------------------------------------
1               0         0                no       no
3                2          2                yes        no
```

Syntax: show dot1x mac-session brief

The following table describes the information displayed by the **show dot1x mac-session brief** command.

Table 42.7: Output from the show dot1x mac-session brief command

This Field...	Displays...
Port	Information about the users connected to each port.
Number of users	The number of users connected to the port.
Number of Authorized users	The number of users connected to the port that have been successfully authenticated.
Dynamic VLAN	Whether the port is a member of a RADIUS-specified VLAN.
Dynamic Filters	Whether RADIUS-specified IP ACLs or MAC address filters have been applied to the port.

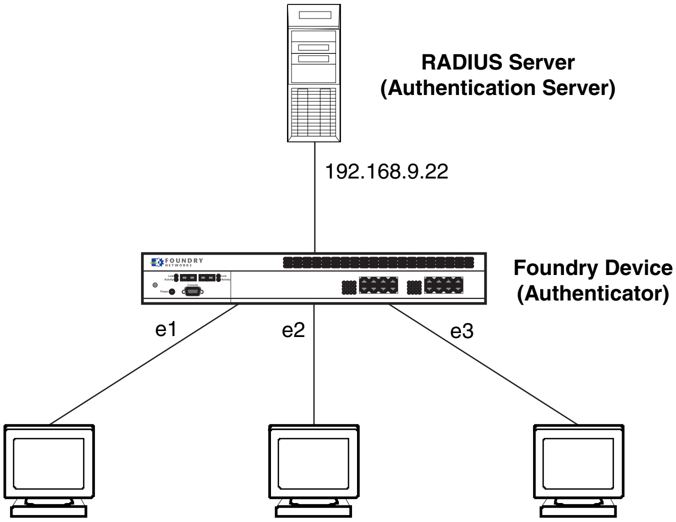
Sample 802.1X Configurations

This section illustrates a sample point-to-point configuration and a sample hub configuration that use 802.1X port security.

Point-to-Point Configuration

Figure 42.6 illustrates a sample 802.1X configuration with Clients connected to three ports on the Foundry device. In a point-to-point configuration, only one 802.1X Client can be connected to each port.

Figure 42.6 Sample Point-to-Point 802.1X Configuration



Clients/Supplicants running 802.1X-compliant client software

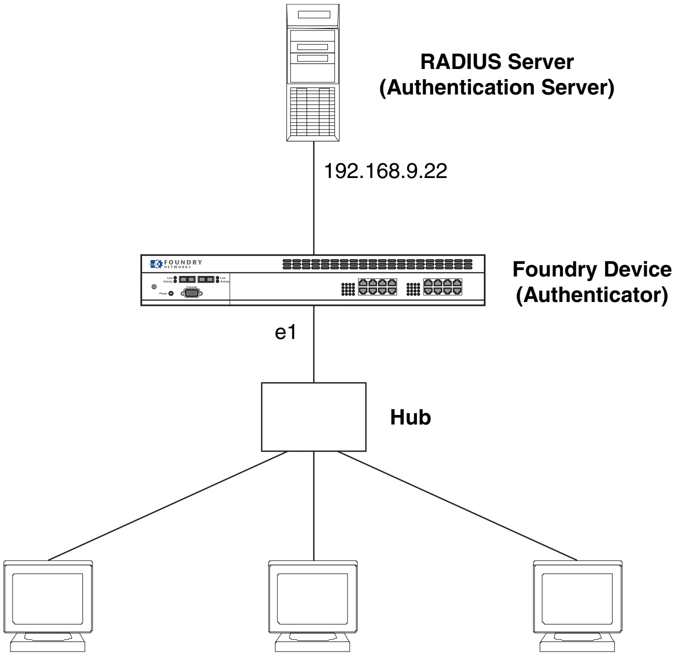
The following commands configure the Foundry device in Figure 42.6:

```
FastIron(config)#aaa authentication dot1x default radius
FastIron(config)#radius-server host 192.168.9.22 auth-port 1812 acct-port 1813
default key mirabeau dot1x

FastIron(config)#dot1x-enable e 1 to 3
FastIron(config-dot1x)#re-authentication
FastIron(config-dot1x)#timeout re-authperiod 2000
FastIron(config-dot1x)#timeout quiet-period 30
FastIron(config-dot1x)#timeout tx-period 60
FastIron(config-dot1x)#max-req 6
FastIron(config-dot1x)#exit

FastIron(config)#interface e 1
FastIron(config-if-e1000-1)#dot1x port-control auto
FastIron(config-if-e1000-1)#exit

FastIron(config)#interface e 2
FastIronconfig-if-e1000-2)#dot1x port-control auto
FastIron(config-if-e1000-2)#exit

FastIron(config)#interface e 3
FastIron(config-if-e1000-3)#dot1x port-control auto
FastIron(config-if-e1000-3)#exit
```

Hub Configuration

Figure 42.7 illustrates a configuration where three 802.1X-enabled Clients are connected to a hub, which is connected to a port on the Foundry device. The configuration is similar to that in Figure 42.6, except that 802.1X port security is enabled on only one port, and the **multiple-hosts** command is used to allow multiple Clients on the port.

Figure 42.7 Sample 802.1X Configuration Using a Hub



Clients/Supplicants running 802.1X-compliant client software

The following commands configure the Foundry device in Figure 42.7:

```
FastIron(config)#aaa authentication dot1x default radius
FastIron(config)#radius-server host 192.168.9.22 auth-port 1812 acct-port 1813
default key mirabeau dot1x

FastIron(config)#dot1x-enable e 1
FastIron(config-dot1x)#re-authentication
FastIron(config-dot1x)#timeout re-authperiod 2000
FastIron(config-dot1x)#timeout quiet-period 30
FastIron(config-dot1x)#timeout tx-period 60
FastIron(config-dot1x)#max-req 6
FastIron(config-dot1x)#exit

FastIron(config)#interface e 1
FastIron(config-if-e1000-1)#dot1x port-control auto
FastIron(config-if-e1000-1)#dot1x multiple-hosts
FastIron(config-if-e1000-1)#exit
```

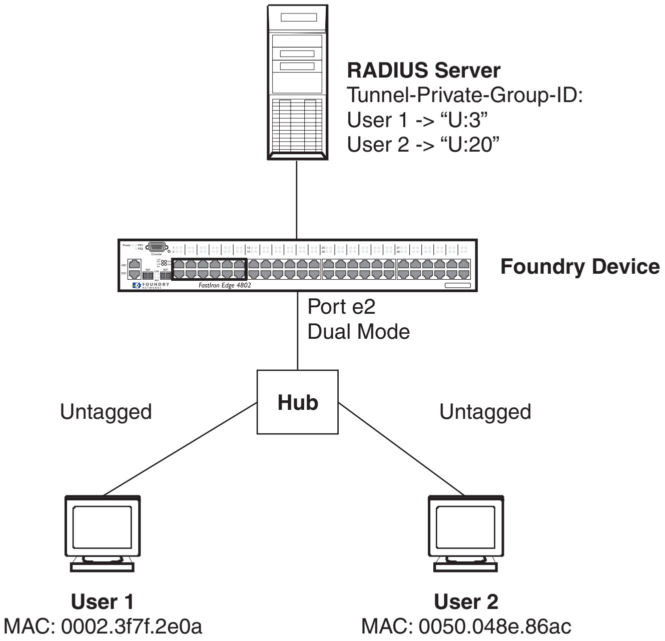
802.1X Authentication with Dynamic VLAN Assignment

Platform Support:

- FESX/FSX/FWSX devices running software release 02.2.00 and later

Figure 42.8 illustrates 802.1X authentication with dynamic VLAN assignment. In this configuration, two users' PCs are connected to a hub, which is connected to port e2. Port e2 is configured as a dual-mode port. Both PCs transmit untagged traffic. The profile for User 1 on the RADIUS server specifies that User 1's PC should be dynamically assigned to VLAN 3. The RADIUS profile for User 2 on the RADIUS server specifies that User 2's PC should be dynamically assigned to VLAN 20.

Figure 42.8 Sample Configuration Using 802.1X Authentication with Dynamic VLAN Assignment



In this example, the PVID for port e2 would be changed based on the first host to be successfully authenticated. If User 1 is authenticated first, then the PVID for port e2 is changed to VLAN 3. If User 2 is authenticated first, then the PVID for port e2 is changed to VLAN 20. Since a PVID cannot be changed by RADIUS authentication after it has been dynamically assigned, if User 2 is authenticated after the port’s PVID was changed to VLAN 3, then User 2 would not be able to gain access to the network.

If there were only one device connected to the port, and authentication failed for that device, it could be placed into the restricted VLAN, where it could gain access to the network.

The part of the running-config related to 802.1X authentication would be as follows:

```
dot1x-enable
 re-authentication
 servertimeout 10
 timeout re-authperiod 10
 auth-fail-action restricted-vlan
 auth-fail-vlanid 1023
 mac-session-aging no-aging permitted-mac-only
 enable ethe 2 to 4
!
!
!
interface ethernet 2
 dot1x port-control auto
 dual-mode
```

If User 1 is successfully authenticated before User 2, the PVID for port e2 would be changed from the default VLAN to VLAN 3.

Had User 2 been the first to be successfully authenticated, the PVID would be changed to 20, and User 1 would not be able to gain access to the network. If there were only one device connected to the port that was sending untagged traffic, and 802.1X authentication failed for that device, it would be placed in the restricted VLAN 1023, and would be able to gain access to the network.

Using Multi-Device Port Authentication and 802.1X Security on the Same Port

Platform Support:

- FESX and FWSX devices running software release 02.2.00 and later

On FESX and FWSX devices running releases 02.2.00 and later, you can configure the Foundry device to use multi-device port authentication and 802.1X security on the same port.

- The multi-device port authentication feature allows you to configure a Foundry device to forward or block traffic from a MAC address based on information received from a RADIUS server. Incoming traffic originating from a given MAC address is switched or forwarded by the device only if the source MAC address is successfully authenticated by a RADIUS server. The MAC address itself is used as the username and password for RADIUS authentication. A connecting user does not need to provide a specific username and password to gain access to the network.
- The IEEE 802.1X standard is a means for authenticating devices attached to LAN ports. Using 802.1X port security, you can configure a Foundry device to grant access to a port based on information supplied by a client to an authentication server.

For information on configuring the multi-device port authentication feature and 802.1X security on Foundry devices, see the related chapters in this book.

When both of these features are enabled on the same port, multi-device port authentication is performed prior to 802.1X authentication. If multi-device port authentication is successful, 802.1X authentication may be performed, based on the configuration of a vendor-specific attribute (VSA) in the profile for the MAC address on the RADIUS server.

When both features are configured on a port, a device connected to the port is authenticated as follows:

1. Multi-device port authentication is performed on the device to authenticate the device’s MAC address.
2. If multi-device port authentication is successful for the device, then the Foundry device checks whether the RADIUS server included the Foundry-802_1x-enable VSA (described in Table 42.8) in the Access-Accept message that authenticated the device.
3. If the Foundry-802_1x-enable VSA is not present in the Access-Accept message, or is present and set to 1, then 802.1X authentication is performed for the device.
4. If the Foundry-802_1x-enable VSA is present in the Access-Accept message, and is set to 0, then 802.1X authentication is skipped. The device is authenticated, and any dynamic VLANs specified in the Access-Accept message returned during multi-device port authentication are applied to the port.
5. If 802.1X authentication is performed on the device, and is successful, then dynamic VLANs or ACLs specified in the Access-Accept message returned during 802.1X authentication are applied to the port.

If multi-device port authentication fails for a device, then by default traffic from the device is either blocked in hardware, or the device is placed in a restricted VLAN. You can optionally configure the Foundry device to perform 802.1X authentication on a device when it fails multi-device port authentication. See “Example 2” on page 42-42 for a sample configuration where this is used.

Configuring Foundry-Specific Attributes on the RADIUS Server

If the RADIUS authentication process is successful, the RADIUS server sends an Access-Accept message to the Foundry device, authenticating the device. The Access-Accept message can include Vendor-Specific Attributes (VSAs) that specify additional information about the device. If you are configuring multi-device port authentication and 802.1X authentication on the same port, then you can configure the Foundry VSAs listed in Table 42.8 on the RADIUS server.

Add these Foundry vendor-specific attributes to your RADIUS server's configuration, and configure the attributes in the individual or group profiles of the devices that will be authenticated. Foundry's Vendor-ID is 1991, with Vendor-Type 1.

Table 42.8: Foundry vendor-specific attributes for RADIUS

Attribute Name	Attribute ID	Data Type	Description
Foundry-802_1x-enable	6	integer	Specifies whether 802.1X authentication is performed when multi-device port authentication is successful for a device. This attribute can be set to one of the following: **0** Do not perform 802.1X authentication on a device that passes multi-device port authentication. Set the attribute to zero for devices that do not support 802.1X authentication. **1** Perform 802.1X authentication when a device passes multi-device port authentication. Set the attribute to one for devices that support 802.1X authentication.
Foundry-802_1x-valid	7	integer	Specifies whether the RADIUS record is valid only for multi-device port authentication, or for both multi-device port authentication and 802.1X authentication. This attribute can be set to one of the following: **0** The RADIUS record is valid only for multi-device port authentication. Set this attribute to zero to prevent a user from using their MAC address as username and password for 802.1X authentication **1** The RADIUS record is valid for both multi-device port authentication and 802.1X authentication.

If neither of these VSAs exist in a device's profile on the RADIUS server, then by default the device is subject to multi-device port authentication (if configured), then 802.1X authentication (if configured). The RADIUS record can be used for both multi-device port authentication and 802.1X authentication.

Example Configurations

The following examples show configurations that use multi-device port authentication and 802.1X authentication on the same port.

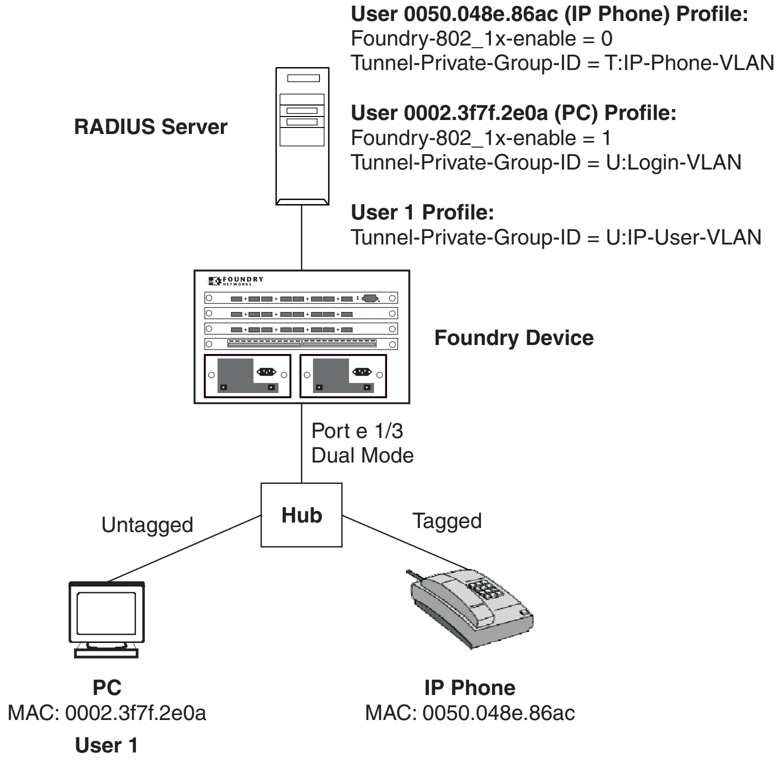
Example 1

Figure 42.9 illustrates an example configuration that uses multi-device port authentication and 802.1X authentication on the same port. In this configuration, a PC and an IP phone are connected to port e 1/3 on a Foundry device. Port e 1/3 is configured as a dual-mode port.

The profile for the PC's MAC address on the RADIUS server specifies that the PC should be dynamically assigned to VLAN "Login-VLAN", and the RADIUS profile for the IP phone specifies that it should be dynamically assigned to the VLAN named "IP-Phone-VLAN". When User 1 is successfully authenticated using 802.1X authentication, the PC is then placed in the VLAN named "User-VLAN".

NOTE: This example assumes that the IP phone initially transmits untagged packets (for example, CDP or DHCP packets), which trigger the authentication process on the Foundry device and client lookup on the RADIUS server. If the phone sends only tagged packets and the port (e 1/3) is not a member of that VLAN, authentication would not occur. In this case, port e 1/3 must be added to that VLAN prior to authentication.

Figure 42.9 Multi-Device Port Authentication and 802.1X Authentication on the Same Port


User 0050.048e.86ac (IP Phone) Profile:
Foundry-802_1x-enable = 0
Tunnel-Private-Group-ID = T:IP-Phone-VLAN
User 0002.3f7f.2e0a (PC) Profile:
Foundry-802_1x-enable = 1
Tunnel-Private-Group-ID = U:Login-VLAN
User 1 Profile:
Tunnel-Private-Group-ID = U:IP-User-VLAN
RADIUS Server
FOUNDRY NETWORKS
Foundry Device
Port e 1/3
Dual Mode
Hub
Untagged
Tagged
PC
MAC: 0002.3f7f.2e0a
User 1
IP Phone
MAC: 0050.048e.86ac

When the devices attempt to connect to the network, they are first subject to multi-device port authentication.

When the IP phone's MAC address is authenticated, the Access-Accept message from the RADIUS server specifies that the IP phone's port be placed into the VLAN named "IP-Phone-VLAN". which is VLAN 7. The Foundry-802_1x-enable attribute is set to 0, meaning that 802.1X authentication is skipped for this MAC address. Port 1/3 is placed in VLAN 7 as a tagged port. No further authentication is performed.

When the PC's MAC address is authenticated, the Access-Accept message from the RADIUS server specifies that the PVID for the PC's port be changed to the VLAN named "Login-VLAN", which is VLAN 1024. The Foundry-802_1x-enable attribute is set to 1, meaning that 802.1X authentication is required for this MAC address. The PVID of the port 1/3 is temporarily changed to VLAN 1024, pending 802.1X authentication.

When User 1 attempts to connect to the network from the PC, he is subject to 802.1X authentication. If User 1 is successfully authenticated, the Access-Accept message from the RADIUS server specifies that the PVID for User 1's port be changed to the VLAN named "User-VLAN", which is VLAN 3. If 802.1X authentication for User 1 is unsuccessful, the PVID for port 1/3 is changed to that of the restricted VLAN, which is 1023, or untagged traffic from port e 1/3 can be blocked in hardware.

The part of the running-config related to port e 1/3 would be as follows:

```
interface ethernet 1/3
```

```
dot1x port-control auto
mac-authentication enable
dual-mode
```

When the PC is authenticated using multi-device port authentication, the port's PVID is changed to "Login-VLAN", which is VLAN 1024 in this example.

When User 1 is authenticated using 802.1X authentication, the port's PVID is changed to "User-VLAN", which is VLAN 3 in this example.

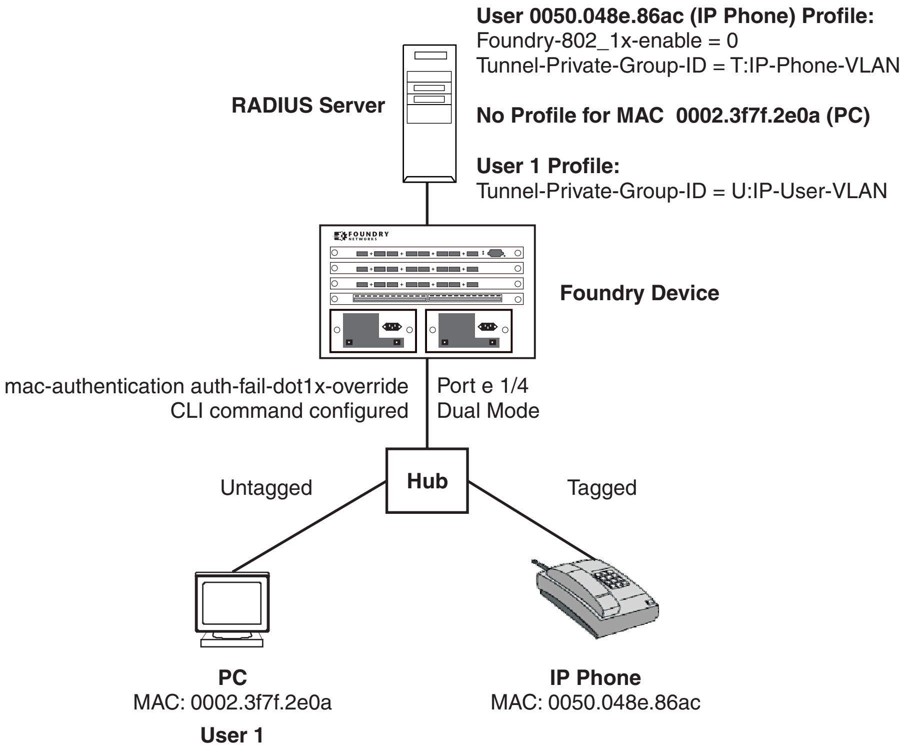
Example 2

The configuration in Figure 42.9 requires that you create a profile on the RADIUS server for each MAC address to which a device or user can connect to the network. In a large network, this can be difficult to implement and maintain.

As an alternative, you can create MAC address profiles only for those devices that do not support 802.1X authentication, such as IP phones and printers, and configure the Foundry device to perform 802.1X authentication for the other devices that do not have MAC address profiles, such as user PCs. To do this, you configure the Foundry device to perform 802.1X authentication when a device fails multi-device port authentication.

Figure 42.10 shows a configuration where multi-device port authentication is performed for an IP phone, and 802.1X authentication is performed for a user's PC. There is a profile on the RADIUS server for the IP phone's MAC address, but not for the PC's MAC address.

Figure 42.10 802.1X Authentication is Performed when a Device Fails Multi-Device Port Authentication


User 0050.048e.86ac (IP Phone) Profile:
Foundry-802_1x-enable = 0
Tunnel-Private-Group-ID = T:IP-Phone-VLAN
RADIUS Server
No Profile for MAC 0002.3f7f.2e0a (PC)
User 1 Profile:
Tunnel-Private-Group-ID = U:IP-User-VLAN
Foundry Device
mac-authentication auth-fail-dot1x-override
CLI command configured
Port e 1/4
Dual Mode
Untagged
Hub
Tagged
PC
MAC: 0002.3f7f.2e0a
User 1
IP Phone
MAC: 0050.048e.86ac

Multi-device port authentication is initially performed for both devices. The IP phone's MAC address has a profile on the RADIUS server. This profile indicates that 802.1X authentication should be skipped for this device, and that the device's port be placed into the VLAN named "IP-Phone-VLAN".

Since there is no profile for the PC's MAC address on the RADIUS server, multi-device port authentication for this MAC address fails. Ordinarily, this would mean that the PVID for the port would be changed to that of the restricted VLAN, or traffic from this MAC would be blocked in hardware. However, the Foundry device is configured to perform 802.1X authentication when a device fails multi-device port authentication, so when User 1 attempts to connect to the network from the PC, he is subject to 802.1X authentication. If User 1 is successfully authenticated, the PVID for port e 1/4 is changed to the VLAN named “User-VLAN”.

NOTE: This example assumes that the IP phone initially transmits untagged packets (for example, CDP or DHCP packets), which trigger the authentication process on the Foundry device and client lookup on the RADIUS server. If the phone sends only tagged packets and the port (e 1/4) is not a member of that VLAN, authentication would not occur. In this case, port e 1/4 must be added to that VLAN prior to authentication.

To configure the Foundry device to perform 802.1X authentication when a device fails multi-device port authentication, enter the following command:

```
FastIron(config)#mac-authentication auth-fail-dot1x-override
```

Syntax: [no] mac-authentication auth-fail-dot1x-override

Chapter 43

Using the MAC Port Security Feature

This chapter describes how to configure Foundry devices to learn “secure” MAC addresses on an inteface so that the interface will forward only packets that match the secure addresses.

Overview

You can configure the Foundry device to learn “secure” MAC addresses on an interface. The interface will forward only packets with source MAC addresses that match these learned secure addresses. The secure MAC addresses can be specified manually, or the Foundry device can learn them automatically. After the device reaches the limit for the number of secure MAC addresses it can learn on the interface, if the interface then receives a packet with a source MAC address that does not match the learned addresses, it is considered a security violation.

When a security violation occurs, a Syslog entry and an SNMP trap are generated. In addition, the device takes one of two actions: either drops packets from the violating address (and allows packets from the secure addresses), or disables the port for a specified amount of time. You specify which of these actions takes place.

The secure MAC addresses are not flushed when an interface is disabled and re-enabled. The secure addresses can be kept secure permanently (the default), or can be configured to age out, at which time they are no longer secure. You can configure the device to automatically save the secure MAC address list to the startup-config file at specified intervals, allowing addresses to be kept secure across system restarts.

The port security feature applies only to Ethernet interfaces.

NOTE: MAC Port Security cannot be configured on 802.1X Port Security-enabled ports.

Local and Global Resources

The port security feature uses a concept of local and global “resources” to determine how many MAC addresses can be secured on each interface. In this context, a “resource” is the ability to store one secure MAC address entry. Each interface is allocated 64 local resources. Additional global resources are shared among all interfaces on the device.

When the port security feature is enabled on an interface, the interface can store one secure MAC address. You can increase the number of MAC addresses that can be secured using local resources to a maximum of 64.

Besides the maximum of 64 local resources available to an interface, there are additional global resources. Depending on flash memory size, a device can have 1024, 2048, or 4096 global resources available. When an interface has secured enough MAC addresses to reach its limit for local resources, it can secure additional MAC addresses by using global resources. Global resources are shared among all the interfaces on a first-come, first-served basis.

The maximum number of MAC addresses any single interface can secure is 64 (the maximum number of local resources available to the interface), plus the number of global resources not allocated to other interfaces.

Configuration Notes and Limitations

The following MAC port security notes and limitations apply to the Foundry devices:

- The MAC port security feature is not supported on static trunk group members or ports that are configured for link aggregation.
- FastIron devices do not support the **reserved-vlan-id <num>** command, which changes the default VLAN ID for the MAC port security feature.
- The SNMP trap generated for restricted MAC addresses indicates the VLAN ID associated with the MAC address, as well as the port number and MAC address.
- You cannot enable MAC port security on the same port that has multi-device port authentication enabled.

Configuring the MAC Port Security Feature

To configure the MAC port security feature, perform the following tasks:

- Enable the MAC port security feature
- Set the maximum number of secure MAC addresses for an interface
- Set the port security age timer
- Specify secure MAC addresses
- Configure the device to automatically save secure MAC addresses to the startup-config file
- Specify the action taken when a security violation occurs
- Deny specific MAC addresses

Enabling the MAC Port Security Feature

By default, the MAC port security feature is disabled on all interfaces. You can enable or disable the feature globally on all interfaces at once or on individual interfaces.

To enable the feature on all interfaces at once:

```
FastIron(config)#port security
FastIron(config-port-security)#enable
```

To disable the feature on all interfaces at once:

```
FastIron(config)#port security
FastIron(config-port-security)#no enable
```

To enable the feature on a specific interface:

```
FastIron(config)#int e 7/11
FastIron(config-if-e1000-7/11)#port security
FastIron(config-port-security-e1000-7/11)#enable
```

Syntax: port security

Syntax: [no] enable

Setting the Maximum Number of Secure MAC Addresses for an Interface

When port security is enabled, an interface can store one secure MAC address. You can increase the number of MAC addresses that can be stored to a maximum of 64, plus the total number of global resources available.

For example, to configure interface 7/11 to have a maximum of 10 secure MAC addresses:

```
FastIron(config)#int e 7/11
FastIron(config-if-e1000-7/11)#port security
FastIron(config-if-e1000-7/11)#maximum 10
```

Syntax: maximum <number-of-addresses>

The <number-of-addresses> parameter can be set to a number from 0 – (64 + the total number of global resources available). The total number of global resources is 2048 or 4096, depending on flash memory size. Setting the parameter to 0 prevents any addresses from being learned. The default is 1.

Setting the Port Security Age Timer

By default, learned MAC addresses stay secure indefinitely. You can optionally configure the device to age out secure MAC addresses after a specified amount of time.

To set the port security age timer to 10 minutes on all interfaces:

```
FastIron(config)#port security
FastIron(config-port-security)#age 10
```

To set the port security age timer to 10 minutes on a specific interface:

```
FastIron(config)#int e 7/11
FastIron(config-if-e1000-7/11)#port security
FastIron(config-port-security-e1000-7/11)#age 10
```

Syntax: [no] age <minutes>

The default is 0 (never age out secure MAC addresses).

Specifying Secure MAC Addresses

This section describes how to configure secure MAC addresses on tagged and untagged interfaces.

On an Untagged Interface

To specify a secure MAC address on an untagged interface, enter commands such as the following:

```
FastIron(config)#int e 7/11
FastIron(config-if-e1000-7/11)#port security
FastIron(config-port-security-e1000-7/11)#secure-mac-address 0050.DA18.747C
```

Syntax: [no] secure-mac-address <mac-address>

On a Tagged Interface

When specifying a secure MAC address on a tagged interface, you must also specify the VLAN ID. To do so, enter commands such as the following:

```
FastIron(config)#int e 7/11
FastIron(config-if-e1000-7/11)#port security
FastIron(config-port-security-e1000-7/11)#secure-mac-address 0050.DA18.747C 2
```

Syntax: [no] secure-mac-address <mac-address> <vlan-ID>

NOTE: If MAC port security is enabled on a port and you change the VLAN membership of the port, make sure that you also change the VLAN ID specified in the **secure-mac-address** configuration statement for the port.

When a secure MAC address is applied to a tagged port, the **vlan-id** is generated for both tagged and untagged ports. When you display the configuration, you will see an entry for the secure MAC addresses `secure-mac-address <address> <vlan>.` For example, you may see the following line:

```
secure-mac-address 0000.1111.2222 10
```

This line means that MAC address 0000.1111.2222 on VLAN 10 is a secure MAC address.

Autosaving Secure MAC Addresses to the Startup-Config File

Learned MAC addresses can automatically be saved to the startup-config file at specified intervals. For example, to automatically save learned secure MAC addresses every twenty minutes, enter the following commands:

```
FastIron(config)#port security
FastIron(config-port-security)#autosave 20
```

Syntax: [no] autosave <minutes>

You can specify from 15 – 1440 minutes. By default, secure MAC addresses are not autosaved to the startup-config file.

Specifying the Action Taken when a Security Violation Occurs

A security violation can occur when a user tries to connect to a port where a MAC address is already locked, or the maximum number of secure MAC addresses has been exceeded. When a security violation occurs, an SNMP trap and Syslog message are generated.

You can configure the device to take one of two actions when a security violation occurs: either drop packets from the violating address (and allow packets from secure addresses), or disable the port for a specified time.

Dropping Packets from a Violating Address

To configure the device to drop packets from a violating address and allow packets from secure addresses:

```
FastIron(config)#int e 7/11
FastIron(config-if-e1000-7/11)#port security
FastIron(config-port-security-e1000-7/11)#violation restrict
```

Syntax: violation restrict

NOTE: When the **restrict** option is used, the maximum number of MAC addresses that can be restricted is 128. If the number of violating MAC addresses exceeds this number, the port is shut down. An SNMP trap and the following Syslog message are generated: "Port Security violation restrict limit 128 exceeded on interface ethernet <port_id>". This is followed by a port shutdown Syslog message and trap.

Specifying the Period of Time to Drop Packets from a Violating Address

Specify a number of minutes that the device drops packets from a violating address using commands similar to the following:

```
FastIron(config)#int e 7/11
FastIron(config-if-e1000-7/11)#port security
FastIron(config-port-security-e1000-7/11)#violation restrict 5
```

Syntax: violation restrict <age>

<age> can be from 0 – 1440 minutes. The default is 5 minutes. Specifying 0 drops packets from the violating address permanently.

Aging for restricted MAC addresses is done in software. There can be a worst case inaccuracy of one minute from the specified time.

The restricted MAC addresses are denied in hardware.

Disabling the Port for a Specified Amount of Time

You can configure the device to disable the port for a specified amount of time when a security violation occurs.

To shut down the port for 5 minutes when a security violation occurs:

```
FastIron(config)#int e 7/11
FastIron(config-if-e1000-7/11)#port security
FastIron(config-port-security-e1000-7/11)#violation shutdown 5
```

Syntax: violation shutdown <minutes>

You can specify from 0 – 1440 minutes. Specifying 0 shuts down the port permanently when a security violation occurs.

Clearing Port Security Statistics

You can clear restricted MAC addresses and violation statistics from ports globally (on all ports) or on individual ports.

Clearing Restricted MAC Addresses

To clear all restricted MAC addresses globally, enter the following command:

```
FastIron#clear port security restricted-macs all
```

To clear restricted MAC addresses on a specific port, enter a command such as the following:

```
FastIron#clear port security restricted-macs e 5
```

Syntax: clear port security restricted-macs all | ethernet [<slot-num>/]<port-num>

The <slotnum> parameter is required on chassis devices.

The <portnum> parameter is a valid port number.

Clearing Violation Statistics

To clear violation statistics globally, enter the following command:

```
FastIron#clear port security statistics all
```

To clear violation statistics on a specific port, enter a command such as the following:

```
FastIron#clear port security statistics e 1/5
```

Syntax: clear port security statistics all | ethernet [<slot-num>/]<port-num>

The <slotnum> parameter is required on chassis devices.

The <portnum> parameter is a valid port number.

Displaying Port Security Information

You can display the following information about the port security feature:

- The port security settings for an individual port or for all the ports on a specified module
- The secure MAC addresses configured on the device
- Port security statistics for an interface or for a module

Displaying Port Security Settings

You can display the port security settings for an individual port or for all the ports on a specified module. For example, to display the port security settings for port 7/11, enter the following command:

```
FastIron#show port security e 7/11
Port  Security Violation Shutdown-Time Age-Time  Max-MAC
----- -------- --------- ------------- --------- -------
 7/11 disabled  shutdown            10        10       1
```

Syntax: show port security ethernet [<slotnum>/]<portnum>

The <slotnum> parameter is required on chassis devices.

The <portnum> parameter is a valid port number.

Table 43.1: Output from the show port security command

This Field...	Displays...
Port	The slot and port number of the interface.
Security	Whether the port security feature has been enabled on the interface.
Violation	The action to be undertaken when a security violation occurs, either “shutdown” or “restrict”.
Shutdown-Time	The number of seconds a port is shut down following a security violation, if the port is set to “shutdown” when a violation occurs.
Age-Time	The amount of time, in minutes, MAC addresses learned on the port will remain secure.
Max-MAC	The maximum number of secure MAC addresses that can be learned on the interface.

Displaying the Secure MAC Addresses

To list the secure MAC addresses configured on the device, enter the following command:

```
FastIron#show port security mac
Port  Num-Addr Secure-Src-Addr Resource Age-Left  Shutdown/Time-Left
----- -------- --------------- -------- --------- ------------------
 7/11        1  0050.da18.747c    Local        10       no
```

Syntax: show port security mac

This command displays the following information:

Table 43.2: Output from the show port security mac command

This Field...	Displays...
Port	The slot and port number of the interface.
Num-Addr	The number of MAC addresses secured on this interface.
Secure-Src-Addr	The secure MAC address.
Resource	Whether the address was secured using a local or global resource. See “Local and Global Resources” on page 43-1 for more information.
Age-Left	The number of minutes the MAC address will remain secure.
Shutdown/Time-Left	Whether the interface has been shut down due to a security violation and the number of seconds before it is enabled again.

Displaying Port Security Statistics

You can display port security statistics for an interface or for a module.

For example, to display port security statistics for interface 7/11:

```
FastIron#show port security statistics e 7/11
Port  Total-Addrs Maximum-Addrs Violation Shutdown/Time-Left
----- ----------- ------------- --------- ------------------
 7/11           1             1         0       no
```

Syntax: show port security statistics [<slotnum>/]<portnum>

The <slotnum> parameter is required on chassis devices.

The <portnum> parameter is a valid port number.

Table 43.3: Output from the show port security statistics <portnum> command

This Field...	Displays...
Port	The slot and port number of the interface.
Total-Addrs	The total number of secure MAC addresses on the interface.
Maximum-Addrs	The maximum number of secure MAC addresses on the interface.
Violation	The number of security violations on the port.
Shutdown/Time-Left	Whether the port has been shut down due to a security violation and the number of seconds before it is enabled again.

To display port security statistics for an interface module, enter the following command:

```
FastIron#show port security statistics 7
Module 7:
  Total ports: 0
  Total MAC address(es): 0
  Total violations: 0
  Total shutdown ports 0
```

Syntax: show port security statistics <module>

Table 43.4: Output from the show port security statistics <module> command

This Field...	Displays...
Total ports:	The number of ports on the module.
Total MAC address(es):	The total number of secure MAC addresses on the module.
Total violations:	The number of security violations encountered on the module.
Total shutdown ports:	The number of ports on the module shut down as a result of security violations.

Displaying Restricted MAC Addresses on a Port

To display a list of restricted MAC addresses on a port, enter a command such as the following:

```
FastIron#show port security e 1/5 restricted-macs
```

Syntax: show port security ethernet [<slotnum>/]<portnum> restricted-macs

The <slotnum> parameter is required on chassis devices.

The <portnum> parameter is a valid port number.

Chapter 44
Configuring Multi-Device Port Authentication

Multi-device port authentication is a way to configure a Foundry device to forward or block traffic from a MAC address based on information received from a RADIUS server.

How Multi-Device Port Authentication Works

The multi-device port authentication feature is a mechanism by which incoming traffic originating from a specific MAC address is switched or forwarded by the device only if the source MAC address is successfully authenticated by a RADIUS server. The MAC address itself is used as the username and password for RADIUS authentication; the user does not need to provide a specific username and password to gain access to the network. If RADIUS authentication for the MAC address is successful, traffic from the MAC address is forwarded in hardware.

If the RADIUS server cannot validate the user's MAC address, then it is considered an authentication failure, and a specified authentication-failure action can be taken. The default authentication-failure action is to drop traffic from the non-authenticated MAC address in hardware. You can also configure the device to move the port on which the non-authenticated MAC address was learned into a restricted or “guest” VLAN, which may have limited access to the network.

RADIUS Authentication

The multi-device port authentication feature communicates with the RADIUS server to authenticate a newly found MAC address. The Foundry device supports multiple RADIUS servers; if communication with one of the RADIUS servers times out, the others are tried in sequential order. If a response from a RADIUS server is not received within a specified time (by default, 3 seconds) the RADIUS session times out, and the device retries the request up to three times. If no response is received, the next RADIUS server is chosen, and the request is sent for authentication.

The RADIUS server is configured with the usernames and passwords of authenticated users. For multi-device port authentication, the username and password is the MAC address itself; that is, the device uses the MAC address for both the username and the password in the request sent to the RADIUS server. For example, given a MAC address of 0007e90feaa1, the users file on the RADIUS server would be configured with a username and password both set to 0007e90feaa1. When traffic from this MAC address is encountered on a MAC-authentication-enabled interface, the device sends the RADIUS server an Access-Request message with 0007e90feaa1 as both the username and password. The format of the MAC address sent to the RADIUS server is configurable through the CLI.

The request for authentication from the RADIUS server is successful only if the username and password provided in the request matches an entry in the users database on the RADIUS server. When this happens, the RADIUS server returns an Access-Accept message back to the Foundry device. When the RADIUS server returns an Access-Accept message for a MAC address, that MAC address is considered authenticated, and traffic from the MAC address is forwarded normally by the Foundry device.

Authentication-Failure Actions

If the MAC address does not match the username and password of an entry in the users database on the RADIUS server, then the RADIUS server returns an Access-Reject message. When this happens, it is considered an authentication failure for the MAC address. When an authentication failure occurs, the Foundry device can either drop traffic from the MAC address in hardware (the default), or move the port on which the traffic was received to a restricted VLAN.

Supported RADIUS Attributes

Foundry FastIron devices support the following RADIUS attributes for multi-device port authentication:

- Username (1) – RFC 2865
- NAS-IP-Address (4) – RFC 2865
- NAS-Port (5) – RFC 2865
- Service-Type (6) – RFC 2865
- FilterId (11) – RFC 2865
- Framed-MTU (12) – RFC 2865
- State (24) – RFC 2865
- Vendor-Specific (26) – RFC 2865
- Session-Timeout (27) – RFC 2865
- Termination-Action (29) – RFC 2865
- Calling-Station-ID (31) – RFC 2865
- NAS-Port-Type (61) – RFC 2865
- Tunnel-Type (64) – RFC 2868
- Tunnel-Medium-Type (65) – RFC 2868
- EAP Message (79) – RFC 2579
- Message-Authenticator (80) RFC 3579
- Tunnel-Private-Group-Id (81) – RFC 2868
- NAS-Port-id (87) – RFC2869

Support for Dynamic VLAN Assignment

Foundry's multi-device port authentication feature supports ***dynamic VLAN assignment***, where a port can be placed in one or more VLANs based on the MAC address learned on that interface. For details about this feature, see "Dynamically Assigning a Port to Multiple VLANs" on page 44-8.

Support for Dynamic ACLs

Foundry's multi-device port authentication implementation supports the assignment of a MAC address to a specific ACL, based on the MAC address learned on the interface. For details about this feature, see "Dynamically Applying IP ACLs to Authenticated MAC Addresses" on page 44-8.

Support for Authenticating Multiple MAC Addresses on an Interface

The multi-device port authentication feature allows multiple MAC addresses to be authenticated or denied authentication on each interface. The maximum number of MAC addresses that can be authenticated on each interface is limited only by the amount of system resources available on the Foundry device.

Using Multi-Device Port Authentication and 802.1X Security on the Same Port

On some Foundry devices, multi-device port authentication and 802.1X security can be configured on the same port.

You can configure the Foundry device to use multi-device port authentication and 802.1X security on the same port. When both of these features are enabled on the same port, multi-device port authentication is performed prior to 802.1X authentication. If multi-device port authentication is successful, 802.1X authentication may be performed, based on the configuration of a vendor-specific attribute (VSA) in the profile for the MAC address on the RADIUS server.

When both features are configured on a port, a device connected to the port is authenticated as follows:

1. Multi-device port authentication is performed on the device to authenticate the device's MAC address.
2. If multi-device port authentication is successful for the device, then the device checks whether the RADIUS server included the Foundry-802_1x-enable VSA (described in Table 44.1) in the Access-Accept message that authenticated the device.
3. If the Foundry-802_1x-enable VSA is not present in the Access-Accept message, or is present and set to 1, then 802.1X authentication is performed for the device.
4. If the Foundry-802_1x-enable VSA is present in the Access-Accept message, and is set to 0, then 802.1X authentication is skipped. The device is authenticated, and any dynamic VLANs specified in the Access-Accept message returned during multi-device port authentication are applied to the port.
5. If 802.1X authentication is performed on the device, and is successful, then dynamic VLANs or ACLs specified in the Access-Accept message returned during 802.1X authentication are applied to the port.

If multi-device port authentication fails for a device, then by default traffic from the device is either blocked in hardware, or the device is placed in a restricted VLAN. You can optionally configure the Foundry device to perform 802.1X authentication on a device when it fails multi-device port authentication. See "Example 2" on page 44-25 for a sample configuration where this is used.

Configuring Foundry-Specific Attributes on the RADIUS Server

If the RADIUS authentication process is successful, the RADIUS server sends an Access-Accept message to the Foundry device, authenticating the device. The Access-Accept message can include Vendor-Specific Attributes (VSAs) that specify additional information about the device. If you are configuring multi-device port authentication and 802.1X authentication on the same port, then you can configure the Foundry VSAs listed in Table 44.1 on the RADIUS server.

You add these Foundry vendor-specific attributes to your RADIUS server's configuration, and configure the attributes in the individual or group profiles of the devices that will be authenticated. Foundry's Vendor-ID is 1991, with Vendor-Type 1.

Table 44.1: Foundry Vendor-Specific Attributes for RADIUS

Attribute Name	Attribute ID	Data Type	Description
Foundry-802_1x-enable	6	integer	Specifies whether 802.1X authentication is performed when multi-device port authentication is successful for a device. This attribute can be set to one of the following: **0** Do not perform 802.1X authentication on a device that passes multi-device port authentication. Set the attribute to zero for devices that do not support 802.1X authentication. **1** Perform 802.1X authentication when a device passes multi-device port authentication. Set the attribute to one for devices that support 802.1X authentication.
Foundry-802_1x-valid	7	integer	Specifies whether the RADIUS record is valid only for multi-device port authentication, or for both multi-device port authentication and 802.1X authentication. This attribute can be set to one of the following: **0** The RADIUS record is valid only for multi-device port authentication. Set this attribute to zero to prevent a user from using their MAC address as username and password for 802.1X authentication **1** The RADIUS record is valid for both multi-device port authentication and 802.1X authentication.

If neither of these VSAs exist in a device profile on the RADIUS server, then by default the device is subject to multi-device port authentication (if configured), then 802.1X authentication (if configured). The RADIUS record can be used for both multi-device port authentication and 802.1X authentication.

An example configuration is shown in “Examples of Multi-Device Port Authentication and 802.1X Authentication Configuration on the Same Port” on page 44-23.

Configuring Multi-Device Port Authentication

Configuring multi-device port authentication on the Foundry device consists of the following tasks:

- Enabling multi-device port authentication globally and on individual interfaces
- Specifying the format of the MAC addresses sent to the RADIUS server (optional)
- Specifying the authentication-failure action (optional)
- Enabling and disabling SNMP traps for multi-device port authentication
- Defining MAC address filters (optional)

- Configuring dynamic VLAN assignment (optional)
- Dynamically assigning a port to multiple VLANs (optional)
- Dynamically Applying IP ACLs to authenticated MAC addresses
- Specifying to which VLAN a port is moved after its RADIUS-specified VLAN assignment expires (optional)
- Saving dynamic VLAN assignments to the running-config file (optional)
- Enabling denial of service attack protection (optional)
- Clearing authenticated MAC addresses (optional)
- Disabling aging for authenticated MAC addresses (optional)
- Configuring the hardware aging period for blocked MAC addresses
- Specifying the aging time for blocked MAC addresses (optional)

Enabling Multi-Device Port Authentication

To enable multi-device port authentication, you first enable the feature globally on the device. On some Foundry devices, you can then enable the feature on individual interfaces.

Globally Enabling Multi-Device Port Authentication

To globally enable multi-device port authentication on the device, enter the following command:

```
FastIron(config)#mac-authentication enable
```

Syntax: [no] mac-authentication enable

Enabling Multi-Device Port Authentication on an Interface

To enable multi-device port authentication on an individual interface, enter a command such as the following:

```
FastIron(config)#mac-authentication enable ethernet 3/1
```

Syntax: [no] mac-authentication enable [<slotnum>/]<portnum> | all

The <slotnum> parameter is required on chassis devices.

The <portnum> parameter is a valid port number.

The **all** option enables the feature on all interfaces at once.

You can enable the feature on an interface at the interface CONFIG level. For example:

```
FastIron(config)#interface e 3/1
FastIron(config-if-e1000-3/1)#mac-authentication enable
```

Syntax: [no] mac-authentication enable

You can also configure multi-device port authentication commands on a range of interfaces. For example:

```
FastIron(config)#int e 3/1 to 3/12
FastIron(config-mif-3/1-3/12)#mac-authentication enable
```

Specifying the Format of the MAC Addresses Sent to the RADIUS Server

When multi-device port authentication is configured, the Foundry device authenticates MAC addresses by sending username and password information to a RADIUS server. The username and password is the MAC address itself; that is, the device uses the MAC address for both the username and the password in the request sent to the RADIUS server.

By default, the MAC address is sent to the RADIUS server in the format *xxxxxxxxxxxx*. You can optionally configure the device to send the MAC address to the RADIUS server in the format *xx-xx-xx-xx-xx-xx*, or the format *xxxx.xxxx.xxxx*. To do this, enter a command such as the following:

```
FastIron(config)#mac-authentication auth-passwd-format xxxx.xxxx.xxxx
```

Syntax: [no] mac-authentication auth-passwd-format xxxx.xxxx.xxxx | xx-xx-xx-xx-xx-xx | xxxxxxxxxxxx

Specifying the Authentication-Failure Action

When RADIUS authentication for a MAC address fails, you can configure the device to perform one of two actions:

- Drop traffic from the MAC address in hardware (the default)
- Move the port on which the traffic was received to a restricted VLAN

To configure the device to move the port to a restricted VLAN when multi-device port authentication fails, enter commands such as the following:

```
FastIron(config)#interface e 3/1
FastIron(config-if-e1000-3/1)#mac-authentication auth-fail-action restrict-vlan 100
```

Syntax: [no] mac-authentication auth-fail-action restrict-vlan [<vlan-id>]

If the ID for the restricted VLAN is not specified at the interface level, the global restricted VLAN ID applies for the interface.

To specify the VLAN ID of the restricted VLAN globally, enter the following command:

```
FastIron(config)#mac-authentication auth-fail-vlan-id 200
```

Syntax: [no] mac-authentication auth-fail-vlan-id <vlan-id>

The command above applies globally to all MAC-authentication-enabled interfaces.

Note that the restricted VLAN must already exist on the device. You cannot configure the restricted VLAN to be a non-existent VLAN. If the port is a tagged or dual-mode port, you cannot use a restricted VLAN as the authentication-failure action.

To configure the device to drop traffic from non-authenticated MAC addresses in hardware, enter commands such as the following:

```
FastIron(config)#interface e 3/1
FastIron(config-if-e1000-3/1)#mac-authentication auth-fail-action block-traffic
```

Syntax: [no] mac-authentication auth-fail-action block-traffic

Dropping traffic from non-authenticated MAC addresses is the default behavior when multi-device port authentication is enabled.

Generating Traps for Multi-Device Port Authentication

You can enable and disable SNMP traps for multi-device port authentication. SNMP traps are enabled by default.

To enable SNMP traps for multi-device port authentication after they have been disabled, enter the following command:

```
FastIron(config)#snmp-server enable traps mac-authentication
```

Syntax: [no] snmp-server enable traps mac-authentication

Use the **no** form of the command to disable SNMP traps for multi-device port authentication.

Defining MAC Address Filters

You can specify MAC addresses that do not have to go through multi-device port authentication. These MAC addresses are considered pre-authenticated, and are not subject to RADIUS authentication. To do this, you can define MAC address filters that specify the MAC addresses to exclude from multi-device port authentication.

You should use a MAC address filter when the RADIUS server itself is connected to an interface where multi-device port authentication is enabled. If a MAC address filter is not defined for the MAC address of the RADIUS server and applied on the interface, the RADIUS authentication process would fail since the device would drop all packets from the RADIUS server itself.

For example, the following command defines a MAC address filter for address 0010.dc58.aca4:

```
FastIron(config)#mac-authentication mac-filter 1 permit 0010.dc58.aca4
```

Syntax: [no] mac-authentication mac-filter <filter>

The following commands apply the MAC address filter on an interface so that address 0010.dc58.aca4 is excluded from multi-device port authentication:

```
FastIron(config)#interface e 3/1
FastIron(config-if-e1000-3/1)#mac-authentication apply-mac-auth-filter 1
```

Syntax: [no] mac-authentication apply-mac-auth-filter <filter-id>

Configuring Dynamic VLAN Assignment

An interface can be dynamically assigned to a VLAN based on the MAC address learned on that interface. When a MAC address is successfully authenticated, the RADIUS server sends the Foundry device a RADIUS Access-Accept message that allows the Foundry device to forward traffic from that MAC address. The RADIUS Access-Accept message can also contain attributes set for the MAC address in its access profile on the RADIUS server.

If one of the attributes in the Access-Accept message specifies a VLAN identifier, and this VLAN is available on the Foundry device, the port is moved from its default VLAN to the specified VLAN.

To enable dynamic VLAN assignment for authenticated MAC addresses, you must add attributes to the profile for the MAC address on the RADIUS server, then enable dynamic VLAN assignment on multi-device port authentication-enabled interfaces. See “Support for Dynamic VLAN Assignment” on page 44-2 for a list of the attributes that must be set on the RADIUS server.

To enable dynamic VLAN assignment on a multi-device port authentication-enabled interface, enter commands such as the following:

```
FastIron(config)#interface e 3/1
FastIron(config-if-e1000-3/1)#mac-authentication enable-dynamic-vlan
```

Syntax: [no] mac-authentication enable-dynamic-vlan

If a previous authentication attempt for a MAC address failed, and as a result the port was placed in the restricted VLAN, but a subsequent authentication attempt was successful, the RADIUS Access-Accept message may specify a VLAN for the port. By default, the Foundry device moves the port out of the restricted VLAN and into the RADIUS-specified VLAN. You can optionally configure the device to leave the port in the the restricted VLAN. To do this, enter the following command:

```
FastIron(config-if-e1000-3/1)#mac-authentication no-override-restrict-vlan
```

When the above command is applied, if the RADIUS-specified VLAN configuration is tagged (e.g., T:1024) and the VLAN is valid, then the port is placed in the RADIUS-specified VLAN as a tagged port and left in the restricted VLAN. If the RADIUS-specified VLAN configuration is untagged (e.g., U:1024), the configuration from the RADIUS server is ignored, and the port is left in the restricted VLAN.

Syntax: [no] mac-authentication no-override-restrict-vlan

Configuration Notes

- If you configure dynamic VLAN assignment on a multi-device port authentication enabled interface, and the Access-Accept message returned by the RADIUS server does not contain a Tunnel-Private-Group-ID attribute, then it is considered an authentication failure, and the configured authentication failure action is performed for the MAC address.
- If the <vlan-name> string does not match either the name or the ID of a VLAN configured on the device, then it is considered an authentication failure, and the configured authentication failure action is performed for the MAC address.
- For tagged or dual-mode ports, if the VLAN ID provided by the RADIUS server does not match the VLAN ID in the tagged packet that contains the authenticated MAC address as its source address, then it is considered an authentication failure, and the configured authentication failure action is performed for the MAC address.
- If an untagged port had previously been assigned to a VLAN through dynamic VLAN assignment, and then another MAC address is authenticated on the same port, but the RADIUS Access-Accept message for the

second MAC address specifies a different VLAN, then it is considered an authentication failure for the second MAC address, and the configured authentication failure action is performed. Note that this applies only if the first MAC address has not yet aged out. If the first MAC address has aged out, then dynamic VLAN assignment would work as expected for the second MAC address.

- For dual mode ports, if the RADIUS server returns **T:<vlan-name>**, the traffic will still be forwarded in the statically assigned PVID. If the RADIUS server returns **U:<vlan-name>**, the traffic will not be forwarded in the statically assigned PVID.

Dynamically Assigning a Port to Multiple VLANs

Platform Support:

- FESX and FWSX devices running software release 02.2.00 and later
- FSX devices running software release 02.3.01 and later

Foundry's multi-device port authentication feature supports dynamic VLAN assignment, where a port can be placed in a VLAN based on the MAC address learned on that interface.

When a MAC address is successfully authenticated, the RADIUS server sends the Foundry device a RADIUS Access-Accept message that allows the device to forward traffic from that MAC address. The RADIUS Access-Accept message can also contain attributes set for the MAC address in its access profile on the RADIUS server. If one of the attributes in the Access-Accept message specifies a VLAN identifier, and this VLAN is available on the Foundry device, the port is moved from its default VLAN to the specified VLAN.

The Foundry device dynamically assigns a port to multiple VLANs, based on the results from the RADIUS server. If the RADIUS Access-Accept message specifies multiple VLAN identifiers, the Foundry device can assign the port to the specified VLANs.

To specify VLAN identifiers on the RADIUS server, add the following attributes to the profile for the MAC address on the RADIUS server, then enable dynamic VLAN assignment on multi-device port authentication-enabled interfaces:

Attribute Name	Type	Value
Tunnel-Type	064	13 (decimal) – VLAN
Tunnel-Medium-Type	065	6 (decimal) – 802
Tunnel-Private-Group-ID	081	<vlan-name> (string) The <vlan-name> value can specify either the name or the number of one or more VLANs configured on the Foundry device.

For information about the attributes, see "Dynamic Multiple VLAN Assignment for 802.1X Ports" on page 42-12.

Also, see the example configuration of "Multi-Device Port Authentication with Dynamic VLAN Assignment" on page 44-22.

Dynamically Applying IP ACLs to Authenticated MAC Addresses

Platform Support:

- FastIron X Series devices running software release 04.0.00 and later – L2, BL3, L3
- FGS and FLS devices running software release 04.1.00 and later

Foundry's multi-device port authentication implementation supports the assignment of a MAC address to a specific ACL, based on the MAC address learned on the interface.

When a MAC address is successfully authenticated, the RADIUS server sends the Foundry device a RADIUS Access-Accept message that allows the Foundry device to forward traffic from that MAC address. The RADIUS

Access-Accept message can also contain, among other attributes, the Filter-ID (type 11) attribute for the MAC address. When the Access-Accept message containing the Filter-ID (type 11) attribute is received by the Foundry device, it will use the information in these attributes to apply an IP ACL on a per-MAC (per user) basis.

The dynamic IP ACL is active as long as the client is connected to the network. When the client disconnects from the network, the IP ACL is no longer applied to the port. If an IP ACL had been applied to the port prior to multi-device port authentication; it will be re-applied to the port.

The Foundry device uses information in the Filter ID to apply an IP ACL on a per-user basis. The Filter-ID attribute can specify the number of an existing IP ACL configured on the Foundry device. If the Filter-ID is an ACL number, the specified IP ACL is applied on a per-user basis.

Configuration Considerations and Guidelines

- Dynamic ACL filters are not supported on virtual Interfaces, tagged ports, and dual-mode ports in the base Layer 3 and full Layer 3 codes.
- Dynamic IP ACLs with multi-device port authentication are supported. Dynamic MAC filters with multi-device port authentication are not supported.
- Dynamic IP ACLs are not supported when **acl-per-port-per-vlan** is enabled on a global-basis.
- The RADIUS Filter ID (type 11) attribute is supported. The Vendor-Specific (type 26) attribute is not supported.
- The dynamic ACL must be an extended ACL. Standard ACLs are not supported.
- Multi-device port authentication and 802.1x can be used together on the same port. However, Foundry does not recommend the use of multi-device port authentication and 802.1X with dynamic ACLs together on the same port. If a single supplicant requires both 802.1x and multi-device port authentication, and if both 802.1x and multi-device port authentication try to install different dynamic ACLs for the same supplicant, the supplicant will fail authentication.
- Dynamically assigned IP ACLs are subject to the same configuration restrictions as non-dynamically assigned IP ACLs.
- Dynamic ACL filters are supported only for the inbound direction. Dynamic outbound ACL filters are not supported.
- Dynamic ACL assignment with multi-device port authentication is not supported in conjunction with any of the following features:
  - IP source guard
  - Rate limiting
  - Protection against ICMP or TCP Denial-of-Service (DoS) attacks
  - Policy-based routing
  - 802.1X dynamic filter

Configuring the RADIUS Server to Support Dynamic IP ACLs

When a port is authenticated using multi-device port authentication, an IP ACL filter that exists in the running-config file on the Foundry device can be dynamically applied to the port. To do this, you configure the Filter-ID (type 11) attribute on the RADIUS server. The Filter-ID attribute specifies the name or number of the Foundry IP ACL.

The following is the syntax for configuring the Filter-ID attribute on the RADIUS server to refer to a Foundry IP ACL:

Value	Description
ip.<number>.in[1]	Applies the specified numbered ACL to the authenticated port in the inbound direction.

Value	Description
ip.<name>.in[1,2]	Applies the specified named ACL to the authenticated port in the inbound direction.

[1] The ACL must be an extended ACL. Standard ACLs are not supported.
[2] The <name> in the Filter ID attribute is case-sensitive

The following table lists examples of values you can assign to the Filter-ID attribute on the RADIUS server to refer to IP ACLs configured on a Foundry device.

Possible Values for the Filter ID attribute on the RADIUS server	ACLs configured on the Foundry device
ip.102.in	access-list 102 permit ip 36.0.0.0 0.255.255.255 any
ip.fdry_filter.in	ip access-list standard fdry_filter permit host 36.48.0.3

Specifying to Which VLAN a Port is Moved after its RADIUS-Specified VLAN Assignment Expires

When a port is dynamically assigned to a VLAN through the authentication of a MAC address, and the MAC session for that address is deleted on the Foundry device, then by default the port is removed from its RADIUS-assigned VLAN and placed back in the VLAN where it was originally assigned.

A port can be removed from its RADIUS-assigned VLAN when any of the following occur:

- The link goes down for the port
- The MAC session is manually deleted with the **mac-authentication clear-mac-session** command
- The MAC address that caused the port to be dynamically assigned to a VLAN ages out

For example, say port 1/1 is currently in VLAN 100, to which it was assigned when MAC address 0007.eaa1.e90f was authenticated by a RADIUS server. The port was originally configured to be in VLAN 111. If the MAC session for address 0007.eaa1.e90f is deleted, then port 1/1 is moved from VLAN 100 back into VLAN 111.

You can optionally specify an alternate VLAN to which to move the port when the MAC session for the address is deleted. For example, to place the port in the restricted VLAN, enter commands such as the following:

```
FastIron(config)#interface e 3/1
FastIron(config-if-e1000-3/1)#mac-auth move-back-to-old-vlan port-restrict-vlan
```

Syntax: [no] mac-authentication move-back-to-old-vlan disable | port-configured-vlan | system-default-vlan

The **disable** keyword disables moving the port back to its original VLAN. The port would stay in its RADIUS-assigned VLAN.

The **port-configured-vlan** keyword removes the port from its RADIUS-assigned VLAN and places it back in the VLAN where it was originally assigned. This is the default.

The **port-restrict-vlan** keyword removes the port from its RADIUS-assigned VLAN and places it in the restricted VLAN.

The **system-default-vlan** keyword removes the port from its RADIUS-assigned VLAN and places it in the DEFAULT-VLAN.

Automatic Removal of Dynamic VLAN Assignments for MAC Authenticated Ports

Platform Support: FastIron X Series devices running software release 04.1.00 and later

By default, the Foundry device removes any association between a port and a dynamically-assigned VLAN when all authenticated MAC sessions for that tagged or untagged VLAN have expired on the port. Thus, RADIUS-specified VLAN assignments are not saved to the device's running-config file. When the **show run** command is issued during a session, dynamically-assigned VLANs are not displayed, although they can be displayed with the **show vlan**, **show auth-mac-addresses detail**, and **show auth-mac-addresses authorized-mac** commands.

You can optionally configure the Foundry device to save the RADIUS-specified VLAN assignments to the device's running-config file. See “Saving Dynamic VLAN Assignments to the Running-Config File” , next.

Saving Dynamic VLAN Assignments to the Running-Config File

By default, dynamic VLAN assignments are not saved to the Foundry device's running-config file. However, you can configure the device to do so by entering the following command:

```
FastIron(config)#mac-authentication save-dynamicvlan-to-config
```

When the above command is applied, dynamic VLAN assignments are saved to the running-config file and are displayed when the **show run** command is issued. Dynamic VLAN assignments can also be displayed with the **show vlan**, **show auth-mac-addresses detail**, and **show auth-mac-addresses authorized-mac** commands.

Syntax: [no] mac-authentication save-dynamicvlan-to-config

Enabling Denial of Service Attack Protection

The Foundry device does not start forwarding traffic from an authenticated MAC address in hardware until the RADIUS server authenticates the MAC address; traffic from the non-authenticated MAC addresses is sent to the CPU. A denial of service (DoS) attack could be launched against the device where a high volume of new source MAC addresses is sent to the device, causing the CPU to be overwhelmed with performing RADIUS authentication for these MAC addresses. In addition, the high CPU usage in such an attack could prevent the RADIUS response from reaching the CPU in time, causing the device to make additional authentication attempts.

To limit the susceptibility of the Foundry device to such attacks, you can configure the device to use multiple RADIUS servers, which can share the load when there are a large number of MAC addresses that need to be authenticated. The Foundry device can run a maximum of 10 RADIUS clients per server and will attempt to authenticate with a new RADIUS server if current one times out.

In addition, you can configure the Foundry device to limit the rate of authentication attempts sent to the RADIUS server. When the multi-device port authentication feature is enabled, it keeps track of the number of RADIUS authentication attempts made per second. When you also enable the DoS protection feature, if the number of RADIUS authentication attempts for MAC addresses learned on an interface per second exceeds a configurable rate (by default 512 authentication attempts per second), the device considers this a possible DoS attack and disables the port. You must then manually re-enable the port.

The DoS protection feature is disabled by default. To enable it on an interface, enter commands such as the following:

```
FastIron(config)#interface e 3/1
FastIron(config-if-e1000-3/1)#mac-authentication dos-protection enable
```

Syntax: [no] mac-authentication dos-protection enable

To specify a maximum rate for RADIUS authentication attempts, enter commands such as the following:

```
FastIron(config)#interface e 3/1
FastIron(config-if-e1000-3/1)#mac-authentication dos-protection mac-limit 256
```

Syntax: [no] mac-authentication dos-protection mac-limit <number>

You can specify a rate from 1 – 65535 authentication attempts per second. The default is a rate of 512 authentication attempts per second.

Clearing Authenticated MAC Addresses

The Foundry device maintains an internal table of the authenticated MAC addresses (viewable with the **show authenticated-mac-address** command). You can clear the contents of the authenticated MAC address table either entirely, or just for the entries learned on a specified interface. In addition, you can clear the MAC session for an address learned on a specific interface.

To clear the entire contents of the authenticated MAC address table, enter the following command:

```
FastIron#clear auth-mac-table
```

Syntax: clear auth-mac-table

To clear the authenticated MAC address table of entries learned on a specified interface, enter a command such as the following:

```
FastIron#clear auth-mac-table e 3/1
```

Syntax: clear auth-mac-table [<slotnum>]/<portnum>

The <slotnum> parameter is required on chassis devices.

The <portnum> parameter is a valid port number.

To clear the MAC session for an address learned on a specific interface, enter commands such as the following:

```
FastIron(config)#interface e 3/1
FastIron(config-if-e1000-3/1)#mac-authentication clear-mac-session 00e0.1234.abd4
```

Syntax: mac-authentication clear-mac-session <mac-address>

This command removes the Layer 2 CAM entry created for the specified MAC address. If the Foundry device receives traffic from the MAC address again, the MAC address is authenticated again.

Disabling Aging for Authenticated MAC Addresses

MAC addresses that have been authenticated or denied by a RADIUS server are aged out if no traffic is received from the MAC address for a certain period of time.

- Authenticated MAC addresses or non-authenticated MAC addresses that have been placed in the restricted VLAN are aged out if no traffic is received from the MAC address over the device's normal MAC aging interval.
- Non-authenticated MAC addresses that are blocked by the device are aged out if no traffic is received from the address over a fixed hardware aging period (70 seconds), plus a configurable software aging period. (See the next section for more information on configuring the software aging period).

You can optionally disable aging for MAC addresses subject to authentication, either for all MAC addresses or for those learned on a specified interface.

Globally Disabling Aging of MAC Addresses

On most devices, you can disable aging for all MAC addresses on all interfaces where multi-device port authentication has been enabled by entering the following command:

```
FastIron(config)#mac-authentication disable-aging
```

Syntax: mac-authentication disable-aging

Enter the command at the global or interface configuration level.

The **denied-only** parameter prevents denied sessions from being aged out, but ages out permitted sessions.

The **permitted-only** parameter prevents permitted (authenticated and restricted) sessions from being aged out and ages denied sessions.

Disabling the Aging of MAC Addresses on Interfaces

To disable aging for all MAC addresses subject to authentication on a specific interface where multi-device port authentication has been enabled, enter the command at the interface level. For example:

```
FastIron(config)#interface e 3/1
FastIron(config-if-e1000-3/1)#mac-authentication disable-aging
```

Syntax: [no] mac-authentication disable-aging

Changing the Hardware Aging Period for Blocked MAC Addresses

The feature was added in release 02.2.00 for the FESX and FWSX, and release 02.3.01 for the FSX.

When the Foundry device is configured to drop traffic from non-authenticated MAC addresses, traffic from the blocked MAC addresses is dropped in hardware, without being sent to the CPU. A Layer 2 hardware entry is created that drops traffic from the MAC address in hardware. If no traffic is received from the MAC address for a certain amount of time, this Layer 2 hardware entry is aged out. If traffic is subsequently received from the MAC address, then an attempt can be made to authenticate the MAC address again.

Aging of the Layer 2 hardware entry for a blocked MAC address occurs in two phases, known as hardware aging and software aging.

On FastIron devices, the hardware aging period for blocked MAC addresses is fixed at 70 seconds and is non-configurable. (The hardware aging time for non-blocked MAC addresses is the length of time specified with the **mac-age** command.) The software aging period for blocked MAC addresses is configurable through the CLI, with the **mac-authentication max-age** command. Once the hardware aging period ends, the software aging period begins. When the software aging period ends, the blocked MAC address ages out, and can be authenticated again if the Foundry device receives traffic from the MAC address.

On FastIron X Series devices, the hardware aging period for blocked MAC addresses is not fixed at 70 seconds. The hardware aging period for blocked MAC addresses is equal to the length of time specified with the **mac-age** command. As on FastIron devices, once the hardware aging period ends, the software aging period begins. When the software aging period ends, the blocked MAC address ages out, and can be authenticated again if the device receives traffic from the MAC address.

To change the hardware aging period for blocked MAC addresses, enter a command such as the following:

```
FastIron(config)#mac-authentication hw-deny-age 10
```

Syntax: [no] mac-authentication hw-deny-age <num>

The <num> parameter is a value from 1 to 65535 seconds. The default is 70 seconds.

Specifying the Aging Time for Blocked MAC Addresses

When the Foundry device is configured to drop traffic from non-authenticated MAC addresses, traffic from the blocked MAC addresses is dropped in hardware, without being sent to the CPU. A Layer 2 CAM entry is created that drops traffic from the blocked MAC address in hardware. If no traffic is received from the blocked MAC address for a certain amount of time, this Layer 2 CAM entry is aged out. If traffic is subsequently received from the MAC address, then an attempt can be made to authenticate the MAC address again.

Aging of the Layer 2 CAM entry for a blocked MAC address occurs in two phases, known as ***hardware aging*** and ***software aging***. The hardware aging period is fixed at 70 seconds and is non-configurable. The software aging time is configurable through the CLI.

Once the Foundry device stops receiving traffic from a blocked MAC address, the hardware aging begins and lasts for a fixed period of time. After the hardware aging period ends, the software aging period begins. The software aging period lasts for a configurable amount of time (by default 120 seconds). After the software aging period ends, the blocked MAC address ages out, and can be authenticated again if the Foundry device receives traffic from the MAC address.

To change the length of the software aging period for blocked MAC addresses, enter a command such as the following:

```
FastIron(config)#mac-authentication max-age 180
```

Syntax: [no] mac-authentication max-age <seconds>

You can specify from 1 – 65535 seconds. The default is 120 seconds.

Specifying the RADIUS Timeout Action

Platform Support:

- FGS and FLS devices running software release 04.1.00 and later
- FSX devices running software release 03.2.00 and later (automatic pass-fail authentication)
- FastIron X Series devices running software release 04.1.00 and later

A RADIUS timeout occurs when the Foundry device does not receive a response from a RADIUS server within a specified time limit and after a certain number of retries. The time limit and number of retries can be manually configured using the CLI commands **radius-server timeout** and **radius-server retransmit**, respectively. If the parameters are not manually configured, the Foundry device applies the default value of three seconds with a maximum of three retries.

In releases prior to FSX software release 03.2.00 and FGS software release 04.1.00, when a RADIUS timeout occurs, the Foundry device will automatically reset the authentication process then retry to authenticate the user.

With FSX software release 03.2.00, and FGS software release 04.1.00, and later, you can better control port behavior when a RADIUS timeout occurs by configuring a port on the Foundry device to automatically pass or fail user authentication. A ***pass*** essentially bypasses the authentication process and permits user access to the network. A ***fail*** bypasses the authentication process and blocks user access to the network, unless **restrict-vlan** is configured, in which case, the user is placed into a VLAN with restricted or limited access. By default, the Foundry device will reset the authentication process and ***retry*** to authenticate the user.

Specify the RADIUS timeout action at the Interface level of the CLI.

Permit User Access to the Network after a RADIUS Timeout

To set the RADIUS timeout behavior to bypass multi-device port authentication and *permit* user access to the network, enter commands such as the following:

```
FastIron(config)#interface ethernet 1/3
FastIron(config-if-e100-1/3)#mac-authentication auth-timeout-action success
```

Syntax: [no] mac-authentication auth-timeout-action success

Once the *success* timeout action is enabled, use the **no** form of the command to reset the RADIUS timeout behavior to *retry*.

Deny User Access to the Network after a RADIUS Timeout

To set the RADIUS timeout behavior to bypass multi-device port authentication and *block* user access to the network, enter commands such as the following:

```
FastIron(config)#interface ethernet 1/3
FastIron(config-if-e100-1/3)#mac-authentication auth-timeout-action failure
```

Syntax: [no] mac-authentication auth-timeout-action failure

Once the *failure* timeout action is enabled, use the **no** form of the command to reset the RADIUS timeout behavior to ***retry***.

NOTE: If **restrict-vlan** is configured along with **auth-timeout-action failure**, the user will be placed into a VLAN with restricted or limited access. See “Allow User Access to a Restricted VLAN after a RADIUS Timeout” .

Allow User Access to a Restricted VLAN after a RADIUS Timeout

To set the RADIUS timeout behavior to bypass multi-device port authentication and place the user in a VLAN with restricted or limited access, enter commands such as the following:

```
FastIron(config)#interface ethernet 1/3
FastIron(config-if-e100-1/3)#mac-authentication auth-fail-action restrict-vlan 100
FastIron(config-if-e100-1/3)#mac-authentication auth-timeout-action failure
```

Syntax: [no] mac-authentication auth-fail-action restrict-vlan [<vlan-id>]

Syntax: [no] mac-authentication auth-timeout-action failure

Multi-Device Port Authentication Password Override

Platform Support:

- FGS and FLS devices running software release 04.1.00 and later
- FESX devices running software release 03.2.00 and later

The multi-device port authentication feature communicates with the RADIUS server to authenticate a newly found MAC address. The RADIUS server is configured with the usernames and passwords of authenticated users. For multi-device port authentication, the username and password is the MAC address itself; that is, the device uses the MAC address for both the username and the password in the request sent to the RADIUS server. For example, given a MAC address of 0007e90feaa1, the users file on the RADIUS server would be configured with a username and password both set to 0007e90feaa1. When traffic from this MAC address is encountered on a MAC-authentication-enabled interface, the device sends the RADIUS server an Access-Request message with 0007e90feaa1 as both the username and password.

In FSX software releases prior to 03.2.00, the MAC address is always used as the username and password for multi-device port authentication. The username and password cannot be changed.

Starting in release 03.2.00, the MAC address becomes the *default* password for multi-device port authentication, and you can optionally configure the device to use a different password. Note that the MAC address is still the username and cannot be changed.

To change the password for multi-device port authentication, enter a command such as the following at the GLOBAL Config Level of the CLI:

```
FastIron(config)#mac-authentication password-override
```

Syntax: [no] mac-authentication password-override <password>

where <password> can have up to 32 alphanumeric characters, but cannot include blank spaces.

Limiting the Number of Authenticated MAC Addresses

You cannot enable MAC port security on the same port that has multi-device port authentication enabled. To simulate the function of MAC port security, you can enter a command such as the following:

```
FastIron(config-if-e1000-2)#mac-authentication max-accepted-session 5
```

Syntax: [no] mac-authentication max-accepted-session <session-number>

This command limits the number of successfully authenticated MAC addresses. Enter a value from 1 - 250 for session-number

Displaying Multi-Device Port Authentication Information

You can display the following information about the multi-device port authentication configuration:

- Information about authenticated MAC addresses
- Information about the multi-device port authentication configuration
- Authentication Information for a specific MAC address or port
- Multi-device port authentication settings and authenticated MAC addresses for each port where the multi-device port authentication feature is enabled
- The MAC addresses that have been successfully authenticated
- The MAC addresses for which authentication was not successful

Displaying Authenticated MAC Address Information

To display information about authenticated MAC addresses on the ports where the multi-device port authentication feature is enabled, enter the following command:

```
FastIron#show auth-mac-address
----------------------------------------------------------------------
Port           Vlan  Accepted MACs    Rejected MACs    Attempted-MACs
----------------------------------------------------------------------
1/18           100   1                  100                0
1/20           40    0                  0                  0
1/22           100   0                  0                  0
4/5            30    0                  0                  0
```

Syntax: show auth-mac-address

The following table describes the information displayed by the **show auth-mac-address** command.

Table 44.2: Output from the show authenticated-mac-address command

This Field...	Displays...
Port	The port number where the multi-device port authentication feature is enabled.
Vlan	The VLAN to which the port has been assigned.
Accepted MACs	The number of MAC addresses that have been successfully authenticated
Rejected MACs	The number of MAC addresses for which authentication has failed.
Attempted-MACs	The rate at which authentication attempts are made for MAC addresses.

Displaying Multi-Device Port Authentication Configuration Information

To display information about the multi-device port authentication configuration, enter the following command:

```
FastIron#show auth-mac-address configuration

Feature enabled                : Yes
Number of Ports enabled        : 4
-----------------------------------------------------------------------
Port  Fail-Action     Fail-vlan   Dyn-vlan  MAC-filter
-----------------------------------------------------------------------
1/18  Block Traffic   1            No        No
1/20  Block Traffic   1            No        No
1/22  Block Traffic   1            No        Yes
4/5   Block Traffic   1            No        No
```

Syntax: show auth-mac-address configuration

The following table describes the output from the **show auth-mac-address configuration** command.

Table 44.3: Output from the show authenticated-mac-address configuration command

This Field...	Displays...
Feature enabled	Whether multi-device port authentication is enabled on the Foundry device.
Number of Ports enabled	The number of ports on which the multi-device port authentication feature is enabled.
Port	Information for each multi-device port authentication-enabled port.
Fail-Action	What happens to traffic from a MAC address for which RADIUS authentication has failed: either block the traffic or assign the MAC address to a restricted VLAN.
Fail-vlan	The restricted VLAN to which non-authenticated MAC addresses are assigned, if the Fail-Action is to assign the MAC address to a restricted VLAN.
Dyn-vlan	Whether RADIUS dynamic VLAN assignment is enabled for the port.
MAC-filter	Whether a MAC filter has been applied to specify pre-authenticated MAC addresses.

Displaying Multi-Device Port Authentication Information for a Specific MAC Address or Port

To display authentication information for a specific MAC address or port, enter a command such as the following:

```
FastIron#show auth-mac-address 0007.e90f.eaa1
-------------------------------------------------------------------------------
MAC/IP Address                  Port        Vlan Authenticated Time  Age CAM
                                                                         Index
-------------------------------------------------------------------------------
0007.e90f.eaa1 : 25.25.25.25    1/18        100  Yes    00d01h10m06s 0   N/A
```

Syntax: show auth-mac-address <mac-address> | <ip-addr> | [<slotnum>/]<portnum>

The <ip-addr> parameter lists the MAC address associated with the specified IP address.

The <slotnum> parameter is required on chassis devices.

The <portnum> parameter is a valid port number.

The following table describes the information displayed by the **show authenticated-mac-address** command for a specified MAC address or port.

Table 44.4: Output from the show authenticated-mac-address <address> command

This Field...	Displays...
MAC/IP Address	The MAC address for which information is displayed. If the packet for which multi-device port authentication was performed also contained an IP address, then the IP address is displayed as well.
Port	The port on which the MAC address was learned.
Vlan	The VLAN to which the MAC address was assigned.
Authenticated	Whether the MAC address was authenticated.

Table 44.4: Output from the show authenticated-mac-address <address> command (Continued)

This Field...	Displays...
Time	The time at which the MAC address was authenticated. If the clock is set on the Foundry device, then the actual date and time are displayed. If the clock has not been set, then the time is displayed relative to when the device was last restarted.
Age	The age of the MAC address entry in the authenticated MAC address list.
CAM Index	If the MAC address is blocked, this is the index entry for the Layer 2 CAM entry created for this MAC address. If the MAC address is not blocked, either through successful authentication or through being placed in the restricted VLAN, then “N/A” is displayed. If the hardware aging period has expired, then “ffff” is displayed for the MAC address during the software aging period.

Displaying the Authenticated MAC Addresses

To display the MAC addresses that have been successfully authenticated, enter the following command:

```
FastIron#show auth-mac-addresses authorized-mac
-------------------------------------------------------------------------------
MAC Address     Port        Vlan Authenticated Time  Age  dot1x
-------------------------------------------------------------------------------
0030.4874.3181   15/23       101   Yes Feb  1 15:56:57 Ena  Ena
000f.ed00.0001   18/1        87    Yes Feb  1 15:57:40 Ena  Ena
000f.ed00.012d   18/1        87    Yes Feb  1 15:57:40 Ena  Ena
000f.ed00.0065   18/1        87    Yes Feb  1 15:57:40 Ena  Ena
000f.ed00.0191   18/1        87    Yes Feb  1 15:57:40 Ena  Ena
000f.ed00.01f5   18/1        87    Yes Feb  1 15:57:40 Ena  Ena
```

Syntax: show auth-mac-addresses authorized-mac

Displaying the Non-Authenticated MAC Addresses

To display the MAC addresses for which authentication was not successful, enter the following command:

```
FastIron#show auth-mac-addresses unauthorized-mac
-------------------------------------------------------------------------------
MAC Address     Port        Vlan Authenticated Time  Age  dot1x
-------------------------------------------------------------------------------
000f.ed00.0321   18/1        87    No  Feb  1 15:57:40 H44  Ena
000f.ed00.0259   18/1        87    No  Feb  1 15:57:40 H44  Ena
000f.ed00.0385   18/1        87    No  Feb  1 15:57:40 H44  Ena
000f.ed00.02bd   18/1        87    No  Feb  1 15:57:40 H44  Ena
000f.ed00.00c9   18/1        87    No  Feb  1 15:57:40 H44  Ena
```

Syntax: show auth-mac-addresses unauthorized-mac

Table 44.5 explains the information in the output.

Displaying Multi-Device Port Authentication Information for a Port

To display a summary of Multi-Device Port Authentication for ports on a device, enter the following command:

```
FastIron#show auth-mac-addresses ethernet 18/1

-------------------------------------------------------------------------------
MAC Address      Port   Vlan Authenticated Time  Age Dot1x
-------------------------------------------------------------------------------
000f.ed00.0001    18/1   87   Yes Feb  1 15:57:40 Ena  Ena
000f.ed00.012d    18/1   87   Yes Feb  1 15:57:40 Ena  Ena
000f.ed00.0321    18/1   87   No  Feb  1 15:57:40 H52  Ena
000f.ed00.0259    18/1   87   No  Feb  1 15:57:40 H52  Ena
000f.ed00.0065    18/1   87   Yes Feb  1 15:57:40 Ena  Ena
000f.ed00.0385    18/1   87   No  Feb  1 15:57:40 H52  Ena
000f.ed00.0191    18/1   87   Yes Feb  1 15:57:40 Ena  Ena
000f.ed00.02bd    18/1   87   No  Feb  1 15:57:40 H52  Ena
000f.ed00.00c9    18/1   87   No  Feb  1 15:57:40 H52  Ena
000f.ed00.01f5    18/1   87   Yes Feb  1 15:57:40 Ena  Ena
```

Syntax: show auth-mac-address ethernet <slotnum>/<portnum>

Table 44.5 explains the information in the output.

Table 44.5: Output of show auth-mac-address

This Field...	Displays...
MAC Address	The MAC addresses learned on the port. If the packet for which multi-device port authentication was performed also contained an IP address, the IP address is also displayed.
Port	ID of the port on which the MAC address was learned.
VLAN	VLAN of which the port is a member.
Authenticated	Whether the MAC address has been authenticated by the RADIUS server.
Time	The time the MAC address was authenticated. If the clock is set on the Foundry device, then the actual date and time are displayed. If the clock has not been set, the time is displayed relative to when the device was last restarted.
Age	The age of the MAC address entry in the authenticated MAC address list.
Dot1x	Indicates if 802.1X authentication is enabled or disabled for the MAC address

Displaying Multi-Device Port Authentication Settings and Authenticated MAC Addresses

To display the multi-device port authentication settings and authenticated MAC addresses for a port where the feature is enabled, enter the following command:

```
FastIron#show auth-mac-addresses detailed ethernet 15/23

Port                                : 15/23
Dynamic-Vlan  Assignment            : Disabled
RADIUS failure action               : Block Traffic
   Failure restrict use dot1x       : No
Override-restrict-vlan              : Yes
Port Default VLAN                   : 101 ( RADIUS assigned: No) (101)
Port Vlan State                     : DEFAULT
802.1x override Dynamic PVID        : YES
     override return to PVID        : 101
Original PVID                       : 101
DOS attack protection               : Disabled
Accepted Mac Addresses              : 1
Rejected Mac Addresses              : 0
Authentication in progress          : 0
Authentication attempts             : 0
RADIUS timeouts                     : 0
RADIUS timeouts action              : Success
MAC Address on PVID                 : 1
MAC Address authorized on PVID      : 1
Aging of MAC-sessions               : Enabled
Port move-back vlan                 : Port-configured-vlan
Max-Age of sw mac session           : 120 seconds
hw age for denied mac               : 70 seconds
MAC Filter  applied                 : No
Dynamic Acl applied                 : No
num Dynamic Tagged Vlan             : 2
Dynamic Tagged Vlan list            : 1025 (1/1) 4060 (1/0)

-------------------------------------------------------------------------------
MAC Address     RADIUS Server  Authenticated  Time  Age  Dot1x

-------------------------------------------------------------------------------
0030.4874.3181 64.12.12.5      Yes Feb  1 15:56:57  Ena  Ena
```

Syntax: show auth-mac-address [detail] [ethernet [<slotnum>/]<portnum>]

The <slotnum> parameter is required on chassis devices.

The <portnum> parameter is a valid port number.

Omitting the <portnum> parameter displays information for all interfaces where the multi-device port authentication feature is enabled.

The following table describes the information displayed by the **show auth-mac-addresses detailed** command.

Table 44.6: Output from the show auth-mac-addresses detailed command

This Field...	Displays...
Port	The port to which this information applies.
Dynamic-Vlan Assignment	Whether RADIUS dynamic VLAN assignment has been enabled for the port.
RADIUS failure action	What happens to traffic from a MAC address for which RADIUS authentication has failed: either block the traffic or assign the MAC address to a restricted VLAN.
Failure restrict use dot1x	Indicates if 802.1x traffic that failed multi-device port authentication, but succeeded 802.1x authentication to gain access to the network.
Override-restrict-vlan	Whether a port can be dynamically assigned to a VLAN specified by a RADIUS server, if the port had been previously placed in the restricted VLAN because a previous attempt at authenticating a MAC address on that port failed.
Port Default Vlan	The VLAN to which the port is assigned, and whether the port had been dynamically assigned to the VLAN by a RADIUS server.
Port VLAN state	Indicates the state of the port VLAN. The State can be one of the following: “Default”, “RADIUS Assigned” or “Restricted”.
802.1X override Dynamic PVID	Indicates if 802.1X can dynamically assign a Port VLAN ID (PVID).
override return to PVID	If a port’s PVID is assigned through the multi-device port authentication feature, and 802.1X authentication subsequently specifies a different PVID, then the PVID specified through 802.1X authentication overrides the PVID specified through multi-device port authentication. This line indicates the PVID the port will use if 802.1X dynamically assigns PVID.
Original PVID	The originally configured (not dynamically assigned) PVID for the port.
DOS attack protection	Whether denial of service attack protection has been enabled for multi-device port authentication, limiting the rate of authentication attempts sent to the RADIUS server.
Accepted Mac Addresses	The number of MAC addresses that have been successfully authenticated.
Rejected Mac Addresses	The number of MAC addresses for which authentication has failed.
Authentication in progress	The number of MAC addresses for which authentication is pending. This is the number of MAC addresses for which an Access-Request message has been sent to the RADIUS server, and for which the RADIUS server has not yet sent an Access-Accept message.
Authentication attempts	The total number of authentication attempts made for MAC addresses on an interface, including pending authentication attempts.
RADIUS timeouts	The number of times the session between the Foundry device and the RADIUS server timed out.
RADIUS timeout action	Action to be taken by the RADIUS server if it times out.
MAC address on the PVID	Number of MAC addresses on the PVID.

Table 44.6: Output from the show auth-mac-addresses detailed command (Continued)

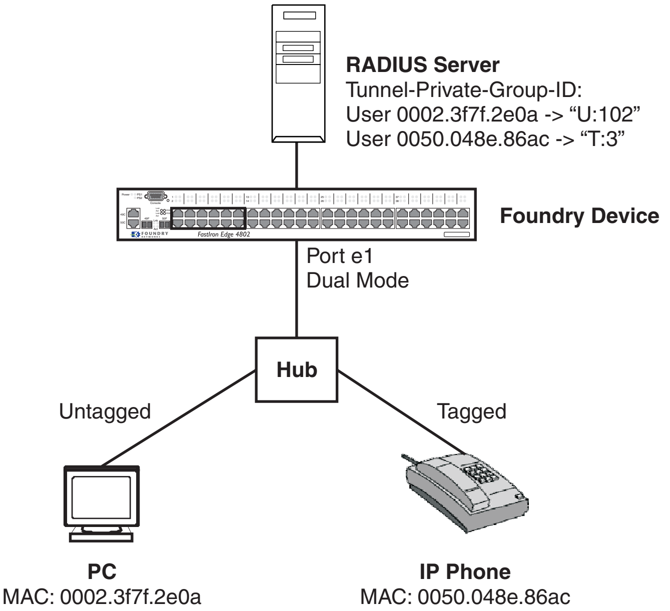
This Field...	Displays...
MAC address authorized on PVID	Number of authorized MAC addresses on the PVID.
Aging of MAC-sessions	Whether software aging of MAC addresses is enabled.
Port move-back VLAN	Indicates the destination VLAN when a RADIUS assigned VLAN is removed. By default, it would return the configured VLAN.
Max-Age of sw MAC-sessions	The configured software aging period for MAC addresses.
hw age for denied MAC	The hardware aging period for blocked MAC addresses. The MAC addresses are dropped in hardware ones the aging period expires.
MAC Filter applied	Indicates whether a MAC filter has been applied to this port to specify pre-authenticated MAC addresses.
Dynamic Acl applied	Indicates whether a dynamic ACL was applied to this port.
num Dynamic Tagged Vlan	The number of dynamically tagged VLANs on this port. **NOTE:** This field was introduced in software releases FSX 04.1.00.
Dynamic Tagged Vlan list	The list of dynamically tagged VLANs on this port. In this example, **1025 (1/1)** indicates that there was one MAC session and one learned MAC address for VLAN 1025. Likewise, **4060 (1/0)** indicates that there was one MAC session and no learned MAC addresses for VLAN 4060. **NOTE:** This field was introduced in software releases FSX 04.1.00.
MAC Address	The MAC addresses learned on the port. If the packet for which multi-device port authentication was performed also contained an IP address, then the IP address is displayed as well.
RADIUS Server	The IP address of the RADIUS server used for authenticating the MAC addresses.
Authenticated	Whether the MAC address has been authenticated by the RADIUS server.
Time	The time at which the MAC address was authenticated. If the clock is set on the Foundry device, then the actual date and time are displayed. If the clock has not been set, then the time is displayed relative to when the device was last restarted.
Age	The age of the MAC address entry in the authenticated MAC address list.
Dot1x	Indicated if 802.1X authentication is enabled or disabled for the MAC address

Example Configurations

Multi-Device Port Authentication with Dynamic VLAN Assignment

Figure 44.1 illustrates multi-device port authentication with dynamic VLAN assignment on a Foundry device. In this configuration, a PC and an IP phone are connected to a hub, which is connected to port e1 on a Foundry device. Port e1 is configured as a dual-mode port. The profile for the PC’s MAC address on the RADIUS server specifies that the PC should be dynamically assigned to VLAN 102, and the RADIUS profile for the IP phone specifies that it should be dynamically assigned to VLAN 3.

Figure 44.1 Using Multi-Device Port Authentication with Dynamic VLAN Assignment



In this example, multi-device port authentication is performed for both devices. If the PC is successfully authenticated, dual-mode port e1's PVID is changed from the VLAN 1 (the DEFAULT-VLAN) to VLAN 102. If authentication for the PC fails, then the PC can be placed in a specified "restricted" VLAN, or traffic from the PC can be blocked in hardware. In this example, if authentication for the PC fails, the PC would be placed in VLAN 1023, the restricted VLAN.

If authentication for the IP phone is successful, then dual-mode port e1 is added to VLAN 3. If authentication for the IP phone fails, then traffic from the IP phone would be blocked in hardware. (Devices sending tagged traffic cannot be placed in the restricted VLAN.)

NOTE: This example assumes that the IP phone initially transmits untagged packets (for example, CDP or DHCP packets), which trigger the authentication process on the Foundry device and client lookup on the RADIUS server. If the phone sends only tagged packets and the port (e1) is not a member of that VLAN, authentication would not occur. In this case, port e1 must be added to that VLAN prior to authentication.

The part of the running-config related to multi-device port authentication would be as follows:

```
mac-authentication enable
mac-authentication auth-fail-vlan-id 1023

interface ethernet 1
 mac-authentication enable
 mac-authentication auth-fail-action restrict-vlan
 mac-authentication enable-dynamic-vlan
 dual-mode
```

Examples of Multi-Device Port Authentication and 802.1X Authentication Configuration on the Same Port

The following examples use multi-device port authentication and 802.1X authentication on the same port.

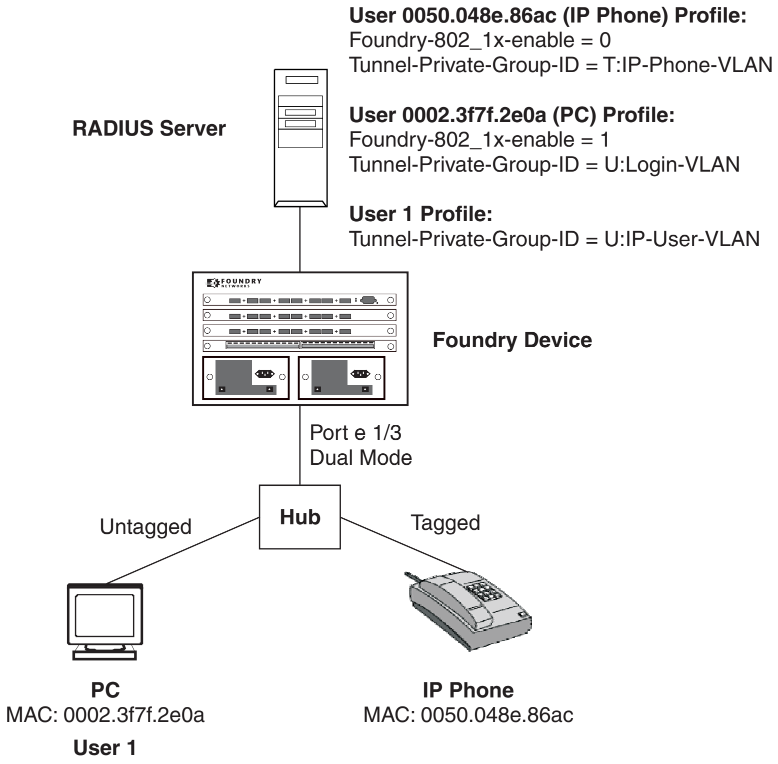
Example 1

Figure 44.2 illustrates an example configuration that uses multi-device port authentication and 802.1X authentication n the same port. In this configuration, a PC and an IP phone are connected to port e 1/3 on an FSX. Port e 1/3 is configured as a dual-mode port.

The profile for the PC's MAC address on the RADIUS server specifies that the PC should be dynamically assigned to VLAN "Login-VLAN", and the profile for the IP phone specifies that it should be dynamically assigned to the VLAN named "IP-Phone-VLAN". When User 1 is successfully authenticated using 802.1X authentication, the PC is then placed in the VLAN named "User-VLAN".

NOTE: This example assumes that the IP phone initially transmits untagged packets (for example, CDP or DHCP packets), which trigger the authentication process on the Foundry device and client lookup on the RADIUS server. If the phone sends only tagged packets and the port (e 1/3) is not a member of that VLAN, authentication would not occur. In this case, port e 1/3 must be added to that VLAN prior to authentication.

Figure 44.2 Using Multi-Device Port Authentication and 802.1X Authentication on the Same Port


User 0050.048e.86ac (IP Phone) Profile:
Foundry-802_1x-enable = 0
Tunnel-Private-Group-ID = T:IP-Phone-VLAN
User 0002.3f7f.2e0a (PC) Profile:
Foundry-802_1x-enable = 1
Tunnel-Private-Group-ID = U:Login-VLAN
User 1 Profile:
Tunnel-Private-Group-ID = U:IP-User-VLAN
RADIUS Server
FOUNDRY NETWORKS
Foundry Device
Port e 1/3
Dual Mode
Hub
Untagged
Tagged
PC
MAC: 0002.3f7f.2e0a
User 1
IP Phone
MAC: 0050.048e.86ac

When the devices attempt to connect to the network, they are first subject to multi-device port authentication.

When the IP phone's MAC address is authenticated, the Access-Accept message from the RADIUS server specifies that the IP phone's port be placed into the VLAN named "IP-Phone-VLAN". which is VLAN 7. The Foundry-802_1x-enable attribute is set to 0, meaning that 802.1X authentication is skipped for this MAC address. Port e 1/3 is placed in VLAN 7 as a tagged port. No further authentication is performed.

When the PC's MAC address is authenticated, the Access-Accept message from the RADIUS server specifies that the PVID for the PC's port be changed to the VLAN named "Login-VLAN", which is VLAN 1024. The Foundry-802_1x-enable attribute is set to 1, meaning that 802.1X authentication is required for this MAC address. The PVID of the port e 1/3 is temporarily changed to VLAN 1024, pending 802.1X authentication.

When User 1 attempts to connect to the network from the PC, he is subject to 802.1X authentication. If User 1 is successfully authenticated, the Access-Accept message from the RADIUS server specifies that the PVID for User

1's port be changed to the VLAN named "User-VLAN", which is VLAN 3. If 802.1X authentication for User 1 is unsuccessful, the PVID for port e 1/3 is changed to that of the restricted VLAN, which is 1023, or untagged traffic from port e 1/3 can be blocked in hardware.

The part of the running-config related to port e 1/3 would be as follows:

```
interface ethernet 1/3
 dot1x port-control auto
 mac-authentication enable
 dual-mode
```

When the PC is authenticated using multi-device port authentication, the port's PVID is changed to "Login-VLAN", which is VLAN 1024 in this example.

When User 1 is authenticated using 802.1X authentication, the port's PVID is changed to "User-VLAN", which is VLAN 3 in this example.

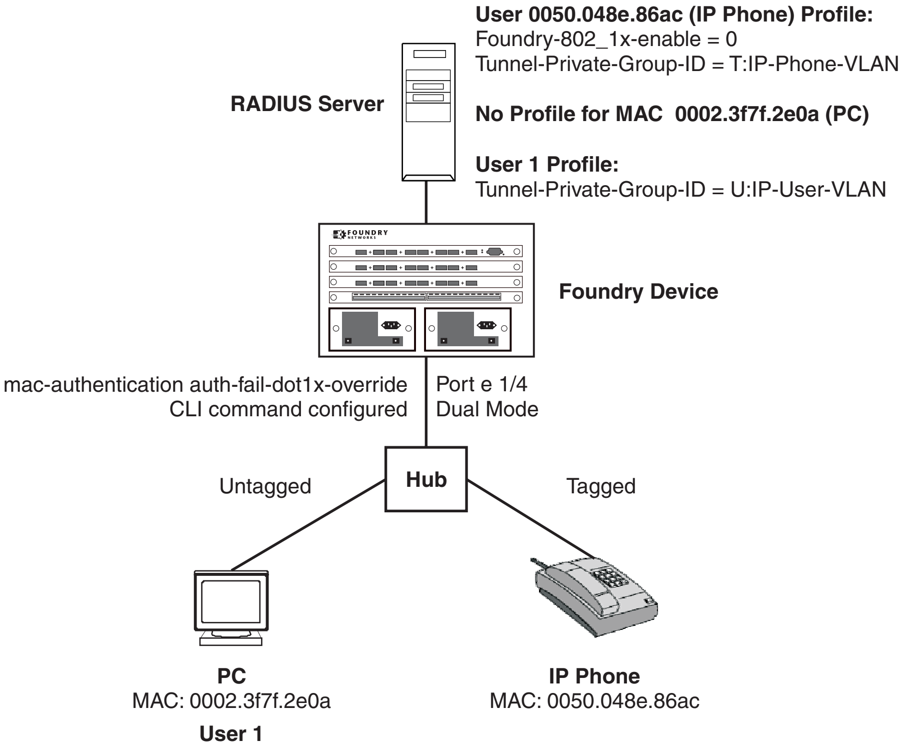
Example 2

The configuration in Figure 44.3 requires that you create a profile on the RADIUS server for each MAC address to which a device or user can connect to the network. In a large network, this can be difficult to implement and maintain.

As an alternative, you can create MAC address profiles only for those devices that do not support 802.1X authentication, such as IP phones and printers, and configure the device to perform 802.1X authentication for the other devices that do not have MAC address profiles, such as user PCs. To do this, you configure the device to perform 802.1X authentication when a device fails multi-device port authentication.

Figure 44.3 shows a configuration where multi-device port authentication is performed for an IP phone, and 802.1X authentication is performed for a user's PC. There is a profile on the RADIUS server for the IP phone's MAC address, but not for the PC's MAC address.

Figure 44.3 802.1X Authentication is Performed When a Device Fails Multi-Device Port Authentication


User 0050.048e.86ac (IP Phone) Profile:
Foundry-802_1x-enable = 0
Tunnel-Private-Group-ID = T:IP-Phone-VLAN
RADIUS Server
No Profile for MAC 0002.3f7f.2e0a (PC)
User 1 Profile:
Tunnel-Private-Group-ID = U:IP-User-VLAN
FOUNDRY NETWORKS
Foundry Device
mac-authentication auth-fail-dot1x-override
CLI command configured
Port e 1/4
Dual Mode
Untagged
Hub
Tagged
PC
MAC: 0002.3f7f.2e0a
User 1
IP Phone
MAC: 0050.048e.86ac

Multi-device port authentication is initially performed for both devices. The IP phone's MAC address has a profile on the RADIUS server. This profile indicates that 802.1X authentication should be skipped for this device, and that the device's port be placed into the VLAN named “IP-Phone-VLAN”.

Since there is no profile for the PC's MAC address on the RADIUS server, multi-device port authentication for this MAC address fails. Ordinarily, this would mean that the PVID for the port would be changed to that of the restricted VLAN, or traffic from this MAC would be blocked in hardware. However, the device is configured to perform 802.1X authentication when a device fails multi-device port authentication, so when User 1 attempts to connect to the network from the PC, he is subject to 802.1X authentication. If User 1 is successfully authenticated, the PVID for port e 1/4 is changed to the VLAN named “User-VLAN”.

NOTE: This example assumes that the IP phone initially transmits untagged packets (for example, CDP or DHCP packets), which trigger the authentication process on the Foundry device and client lookup on the RADIUS server. If the phone sends only tagged packets and the port (e 1/4) is not a member of that VLAN, authentication would not occur. In this case, port e 1/4 must be added to that VLAN prior to authentication.

To configure the device to perform 802.1X authentication when a device fails multi-device port authentication, enter the following command:

```
FastIron(config)#mac-authentication auth-fail-dot1x-override
```

Syntax: [no] mac-authentication auth-fail-dot1x-override

Chapter 45

Protecting Against Denial of Service Attacks

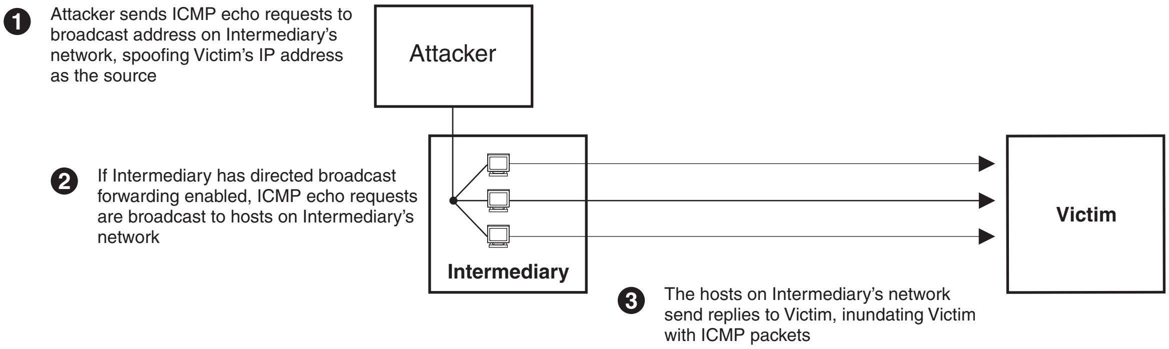
This chapter explains how to protect your Foundry devices from Denial of Service (DoS) attacks.

n a Denial of Service (DoS) attack, a router is flooded with useless packets, hindering normal operation. Foundry devices include measures for defending against two types of DoS attacks: Smurf attacks and TCP SYN attacks.

Protecting Against Smurf Attacks

A ***Smurf attack*** is a kind of DoS attack where an attacker causes a victim to be flooded with ICMP echo (Ping) replies sent from another network. Figure 45.1 illustrates how a Smurf attack works.

Figure 45.1 How a Smurf Attack Floods a Victim with ICMP Replies


❶ Attacker sends ICMP echo requests to broadcast address on Intermediary's network, spoofing Victim's IP address as the source
Attacker
❷ If Intermediary has directed broadcast forwarding enabled, ICMP echo requests are broadcast to hosts on Intermediary's network
Intermediary
Victim
❸ The hosts on Intermediary's network send replies to Victim, inundating Victim with ICMP packets

The attacker sends an ICMP echo request packet to the broadcast address of an intermediary network. The ICMP echo request packet contains the spoofed address of a victim network as its source. When the ICMP echo request reaches the intermediary network, it is converted to a Layer 2 broadcast and sent to the hosts on the intermediary network. The hosts on the intermediary network then send ICMP replies to the victim network.

For each ICMP echo request packet sent by the attacker, a number of ICMP replies equal to the number of hosts on the intermediary network are sent to the victim. If the attacker generates a large volume of ICMP echo request packets, and the intermediary network contains a large number of hosts, the victim can be overwhelmed with ICMP replies.

Avoiding Being an Intermediary in a Smurf Attack

A Smurf attack relies on the intermediary to broadcast ICMP echo request packets to hosts on a target subnet. When the ICMP echo request packet arrives at the target subnet, it is converted to a Layer 2 broadcast and sent to the connected hosts. This conversion takes place only when directed broadcast forwarding is enabled on the device.

To avoid being an intermediary in a Smurf attack, make sure forwarding of directed broadcasts is disabled on the Foundry device. Directed broadcast forwarding is disabled by default. To disable directed broadcast forwarding, do one of the following:

```
FastIron(config)#no ip directed-broadcast
```

Syntax: [no] ip directed-broadcast

Avoiding Being a Victim in a Smurf Attack

You can configure the Foundry device to drop ICMP packets when excessive numbers are encountered, as is the case when the device is the victim of a Smurf attack. You can set threshold values for ICMP packets that are targeted at the router itself or passing through an interface, and drop them when the thresholds are exceeded.

For example, to set threshold values for ICMP packets targeted at the router, enter the following command in CONFIG mode:

```
FastIron(config)#ip icmp burst-normal 5000 burst-max 10000 lockup 300
```

To set threshold values for ICMP packets received on interface 3/11:

```
FastIron(config)#int e 3/11
FastIron(config-if-e1000-3/11)#ip icmp burst-normal 5000 burst-max 10000 lockup 300
```

Syntax: ip icmp burst-normal <value> burst-max <value> lockup <seconds>

The **burst-normal** value can be from 1 – 100000.

The **burst-max** value can be from 1 – 100000.

The **lockup** value can be from 1 – 10000.

This command is supported on Ethernet and Layer 3 ATM interfaces.

The number of incoming ICMP packets per second are measured and compared to the threshold values as follows:

- If the number of ICMP packets exceeds the **burst-normal** value, the excess ICMP packets are dropped.
- If the number of ICMP packets exceeds the **burst-max** value, *all* ICMP packets are dropped for the number of seconds specified by the **lockup** value. When the lockup period expires, the packet counter is reset and measurement is restarted.

In the example above, if the number of ICMP packets received per second exceeds 5,000, the excess packets are dropped. If the number of ICMP packets received per second exceeds 10,000, the device drops all ICMP packets for the next 300 seconds (five minutes).

Protecting Against TCP SYN Attacks

TCP SYN attacks exploit the process of how TCP connections are established in order to disrupt normal traffic flow. When a TCP connection starts, the connecting host first sends a TCP SYN packet to the destination host. The destination host responds with a SYN ACK packet, and the connecting host sends back an ACK packet. This process, known as a “TCP three-way handshake”, establishes the TCP connection.

While waiting for the connecting host to send an ACK packet, the destination host keeps track of the as-yet incomplete TCP connection in a connection queue. When the ACK packet is received, information about the connection is removed from the connection queue. Usually there is not much time between the destination host sending a SYN ACK packet and the source host sending an ACK packet, so the connection queue clears quickly.

In a TCP SYN attack, an attacker floods a host with TCP SYN packets that have random source IP addresses. For each of these TCP SYN packets, the destination host responds with a SYN ACK packet and adds information to the connection queue. However, since the source host does not exist, no ACK packet is sent back to the destination host, and an entry remains in the connection queue until it ages out (after around a minute). If the attacker sends enough TCP SYN packets, the connection queue can fill up, and service can be denied to legitimate TCP connections.

To protect against TCP SYN attacks, you can configure the Foundry device to drop TCP SYN packets when excessive numbers are encountered. You can set threshold values for TCP SYN packets that are targeted at the router itself or passing through an interface, and drop them when the thresholds are exceeded.

For example, to set threshold values for TCP SYN packets targeted at the router, enter the following command in CONFIG mode:

```
FastIron(config)#ip tcp burst-normal 10 burst-max 100 lockup 300
```

To set threshold values for TCP SYN packets received on interface 3/11:

```
FastIron(config)#int e 3/11
FastIron(config-if-e1000-3/11)#ip tcp burst-normal 10 burst-max 100 lockup 300
```

Syntax: ip tcp burst-normal <value> burst-max <value> lockup <seconds>

NOTE: This command is available at the global CONFIG level on both Chassis devices and Compact devices. On Chassis devices, this command is available at the Interface level as well. This command is supported on Ethernet and Layer 3 ATM interfaces.

The **burst-normal** value can be from 1 – 100000.

The **burst-max** value can be from 1 – 100000.

The **lockup** value can be from 1 – 10000.

The number of incoming TCP SYN packets per second are measured and compared to the threshold values as follows:

- If the number of TCP SYN packets exceeds the **burst-normal** value, the excess TCP SYN packets are dropped.
- If the number of TCP SYN packets exceeds the **burst-max** value, *all* TCP SYN packets are dropped for the number of seconds specified by the **lockup** value. When the lockup period expires, the packet counter is reset and measurement is restarted.

In the example above, if the number of TCP SYN packets received per second exceeds 10, the excess packets are dropped. If the number of TCP SYN packets received per second exceeds 100, the device drops all TCP SYN packets for the next 300 seconds (five minutes).

TCP Security Enhancement

TCP security enhancement improves upon the handling of TCP inbound segments. This enhancement eliminates or minimizes the possibility of a TCP reset attack, in which a perpetrator attempts to prematurely terminate an active TCP session, and a data injection attack, wherein an attacker injects or manipulates data in a TCP connection.

In both cases, the attack is blind, meaning the perpetrator does not have visibility into the content of the data stream between two devices, but blindly injects traffic. Also, the attacker does not see the direct effect, the continuing communications between the devices and the impact of the injected packet, but may see the indirect impact of a terminated or corrupted session.

The TCP security enhancement prevents and protects against the following three types of attacks:

- Blind TCP reset attack using the reset (RST) bit.
- Blind TCP reset attack using the synchronization (SYN) bit
- Blind TCP packet injection attack

The TCP security enhancement is automatically enabled.

Protecting Against a Blind TCP Reset Attack Using the RST Bit

In a blind TCP reset attack using the RST bit, a perpetrator attempts to guess the RST segments in order to prematurely terminate an active TCP session.

Prior software releases apply the following rules to the RST bit when receiving TCP segments:

- If the RST bit is set and the sequence number is outside the expected window, the Foundry device silently drops the segment.
- If the RST bit is set and the sequence number is within the acceptable range, the Foundry device resets the connection

To prevent a user from using the RST bit to reset a TCP connection, the RST bit is subject to the following rules when receiving TCP segments:

- If the RST bit is set and the sequence number is outside the expected window, the Foundry device silently drops the segment.
- If the RST bit is exactly the next expected sequence number, the Foundry device resets the connection.
- If the RST bit is set and the sequence number does not exactly match the next expected sequence value, but is within the acceptable window, the Foundry device sends an acknowledgement.

Protecting Against a Blind TCP Reset Attack Using the SYN Bit

In a blind TCP reset attack, a perpetrator attempts to guess the SYN bits to prematurely terminate an active TCP session.

Prior software releases apply the following rules to the SYN bit when receiving TCP segments:

- If the SYN bit is set and the sequence number is outside the expected window, the Foundry device sends an ACK back to the sender.
- If the SYN bit is set and the sequence number is acceptable, the Foundry device sends a RST segment to the peer.

To prevent a user from using the SYN bit to tear down a TCP connection, in current software releases, the SYN bit is subject to the following rules when receiving TCP segments:

- If the SYN bit is set and the sequence number is outside the expected window, the Foundry device sends an acknowledgement (ACK) back to the peer.
- If the SYN bit is set and the sequence number is an exact match to the next expected sequence, the Foundry device sends an ACK segment to the peer. Before sending the ACK segment, the software subtracts one from the value being acknowledged.
- If the SYN bit is set and the sequence number is acceptable, the Foundry device sends an acknowledgement (ACK) segment to the peer.

Protecting Against a Blind Injection Attack

In a blind TCP injection attack, a perpetrator tries to inject or manipulate data in a TCP connection.

To reduce the chances of a blind injection attack, an additional check on all incoming TCP segments is performed.

Displaying Statistics about Packets Dropped Because of DoS Attacks

To display information about ICMP and TCP SYN packets dropped because burst thresholds were exceeded:

```
FastIron#show statistics dos-attack
---------------------------- Local Attack Statistics ---------------------------
ICMP Drop Count    ICMP Block Count    SYN Drop Count    SYN Block Count
---------------    ----------------    --------------    ---------------
              0                   0                 0                  0
--------------------------- Transit Attack Statistics -------------------------
Port   ICMP Drop Count    ICMP Block Count    SYN Drop Count    SYN Block Count
-----  ---------------    ----------------    --------------    ---------------

3/11                 0                   0                 0                  0
```

Syntax: show statistics dos-attack

To clear statistics about ICMP and TCP SYN packets dropped because burst thresholds were exceeded:

```
FastIron#clear statistics dos-attack
```

Syntax: clear statistics dos-attack

Chapter 46
Inspecting and Tracking DHCP Packets

For enhanced network security, you can configure the Foundry device to inspect and keep track of Dynamic Host Configuration Protocol (DHCP) assignments.

Dynamic ARP Inspection

Platform Support:

- FESX/FSX/FWSX devices running software release 03.0.00 and later

Dynamic ARP Inspection (DAI) enables the Foundry device to intercept and examine all ARP request and response packets in a subnet and discard those packets with invalid IP to MAC address bindings. DAI can prevent common man-in-the-middle (MiM) attacks such as ARP cache poisoning, and disallow mis-configuration of client IP addresses.

ARP Poisoning

ARP provides IP communication within a Layer 2 broadcast domain by mapping an IP address to a MAC address. Before a host can talk to another host, it must map the IP address to a MAC address first. If the host does not have the mapping in its ARP table, it creates an ARP request to resolve the mapping. All computers on the subnet will receive and process the ARP requests, and the host whose IP address matches the IP address in the request will send an ARP reply.

An ARP poisoning attack can target hosts, switches, and routers connected to the Layer 2 network by poisoning the ARP caches of systems connected to the subnet and by intercepting traffic intended for other hosts on the subnet. For instance, a malicious host can reply to an ARP request with its own MAC address, thereby causing other hosts on the same subnet to store this information in their ARP tables or replace the existing ARP entry. Furthermore, a host can send gratuitous replies without having received any ARP requests. A malicious host can also send out ARP packets claiming to have an IP address that actually belongs to another host (e.g. the default router). After the attack, all traffic from the device under attack flows through the attacker’s computer and then to the router, switch, or host.

How DAI Works

DAI allows only valid ARP requests and responses to be forwarded.

A Foundry device on which DAI is configured does the following:

- Intercepts ARP packets received by the system CPU
- Inspects all ARP requests and responses received on untrusted ports
- Verifies that each of the intercepted packets has a valid IP-to-MAC address binding before updating the local

ARP table, or before forwarding the packet to the appropriate destination

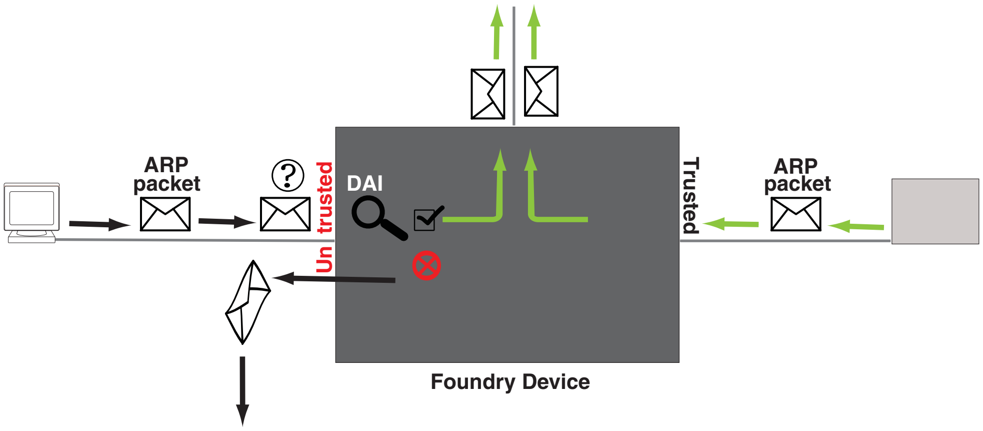
- Drops invalid ARP packets

When you enable DAI on a VLAN, by default, all member ports are untrusted. You must manually configure trusted ports. In a typical network configuration, ports connected to host ports are untrusted. You configure ports connected to other switches or routers as trusted.

DAI inspects ARP packets received on untrusted ports, as shown in Figure 46.1. DAI carries out the inspection based on IP-to-MAC address bindings stored in a trusted binding database. For the Foundry device, the binding database is the ARP table, which supports DAI, DHCP snooping, and IP Source Guard. To inspect an ARP request packet, DAI checks the source IP and source MAC address against the ARP table. For an ARP reply packet, DAI checks the source IP, source MAC, destination IP, and destination MAC addresses. DAI forwards the valid packets and discards those with invalid IP-to-MAC address bindings.

When ARP packets reach a trusted port, DAI lets them through, as shown in Figure 46.1.

Figure 46.1 Dynamic ARP Inspection at Work



ARP Entries

DAI uses the IP/MAC mappings in the ARP table to validate ARP packets received on untrusted ports.

ARP entries in the ARP table derive from the following:

- Dynamic ARP – normal ARP learned from trusted ports.
- Static ARP – statically configured IP/MAC/port mapping.
- Inspection ARP – statically configured IP/MAC mapping, where the port is initially unspecified. The actual physical port mapping will be resolved and updated from validated ARP packets. See “Configuring an Inspection ARP Entry” on page 46-3.
- DHCP-Snooping ARP – information collected from snooping DHCP packets when DHCP snooping is enabled on VLANs.

The status of an ARP entry is either pending or valid:

- Valid – the mapping is valid, and the port is resolved. This is always the case for static ARP entries.
- Pending – for normal dynamic, inspection ARP, and DHCP-Snooping ARP entries before they are resolved, and the port mapped. Their status changes to valid when they are resolved, and the port mapped.

See also: “System Reboot and the Binding Database” on page 46-5.

Configuration Notes and Feature Limitations

The following limits and restrictions apply when configuring DAI:

- Foundry recommends that you do not enable DAI on a trunk port.

- The maximum number of DHCP and static DAI entries depends on the maximum number of ARP table entries allowed on the device. A FastIron switch can have up to 256 ARP entries and a FastIron router can have up to 64,000 ARP entries. In a FastIron router, you can use the **system-max arp** command to change the maximum number of ARP entries for the device.

  However, only up to 1024 DHCP entries can be saved to flash.
- Dynamic ARP entries will be created even if DHCP Snooping is disabled. The router obtains the IP address to MAC address mapping from the DHCP packets.
- Starting in software release FSX 04.0.01, ACLs are supported on member ports of a VLAN on which DHCP snooping and Dynamic ARP Inspection (DAI) are enabled.

Configuring DAI

Configuring DAI consists of the following steps:

1. Configure inspection ARP entries for hosts on untrusted ports. See "Configuring an Inspection ARP Entry" on page 46-3.
2. Enable DAI on a VLAN to inspect ARP packets. See "Enabling DAI on a VLAN" on page 46-3.
3. Configure the trust settings of the VLAN members. ARP packets received on *trusted* ports bypass the DAI validation process. ARP packets received on *untrusted* ports go through the DAI validation process. See "Enabling Trust on a Port" on page 46-4.
4. Enable DHCP snooping to populate the DHCP snooping IP-to-MAC binding database.

The following shows the default settings of DAI:

Feature	Default
Dynamic ARP Inspection	Disabled
Trust setting for ports	Untrusted

Configuring an Inspection ARP Entry

Static ARP and static inspection ARP entries need to be configured for hosts on untrusted ports. Otherwise, when DAI checks ARP packets from these hosts against entries in the ARP table, it will not find any entries for them, and the Foundry device will not allow and learn ARP from an untrusted host.

When the inspection ARP entry is resolved with the correct IP/MAC mapping, its status changes from pending to valid.

To configure an inspection ARP entry, enter commands such as the following:

```
FastIron(config)#arp 20.20.20.12  0001.0002.0003 inspection
```

The commands change the CLI to the interface configuration level for virtual interface 2, assign it an IP address, and then define an inspection ARP entry, mapping a device's IP address 20.20.20.12 with its MAC address 0001.0002.0003.

Syntax: [no] arp <index> <ip-addr> <mac-addr> inspection

The index can be from 1 up to the maximum number of static entries allowed.

The <ip-addr> <mac-addr> parameter specifies a device's IP address and MAC address pairing.

Enabling DAI on a VLAN

DAI is disabled by default. To enable DAI on an existing VLAN, enter the following command:

```
FastIron(config)#ip arp inspection vlan 2
```

The command enables DAI on VLAN 2. ARP packets from untrusted ports in VLAN 2 will undergo DAI inspection.

Syntax: [no] ip arp inspection vlan <vlan-number>

The <vlan-number> variable specifies the ID of a configured VLAN.

Enabling Trust on a Port

The default trust setting for a port is untrusted. For ports that are connected to host ports, leave their trust settings as untrusted.

To enable trust on a port, enter commands such as the following:

```
FastIron(config)#interface ethernet 1/4
FastIron(config-if-e10000-1/4)#arp inspection trust
```

The commands change the CLI to the interface configuration level of port 1/4 and set the trust setting of port 1/4 to trusted.

Syntax: [no] arp inspection trust

Displaying ARP Inspection Status and Ports

To display the ARP inspection status for a VLAN and the trusted/untrusted ports in the VLAN, enter the following command:

```
FastIron#show ip arp inspection vlan 2

IP ARP inspection VLAN 2: Disabled
  Trusted Ports :   ethe 1/4
  Untrusted Ports : ethe 2/1 to 2/3 ethe 4/1 to 4/24 ethe 6/1 to 6/4 ethe 8/1 to
 8/4
```

Syntax: show ip arp inspection [vlan <vlan_id>]

The <vlan_id> variable specifies the ID of a configured VLAN.

Displaying the ARP Table

To display the ARP table, enter the following command:

```
FastIron#show arp

Total number of ARP entries: 2

      IP Address          MAC Address      Type     Age      Port        Status

1     10.43.1.1           0004.80a0.4000   Dynamic  0        mgmt1     Valid

2     10.43.1.78          00e0.8160.6ab1   Dynamic  2        mgmt1     Valid
```

The command displays all ARP entries in the system.

Syntax: syntax: show arp

DHCP Snooping

Platform Support:

- FESX/FSX/FWSX devices running software release 03.0.00 and later

Dynamic Host Configuration Protocol (DHCP) snooping enables the Foundry device to filter untrusted DHCP packets in a subnet. DHCP snooping can ward off MiM attacks, such as a malicious user posing as a DHCP server sending false DHCP server reply packets with the intention of misdirecting other users. DHCP snooping can also stop unauthorized DHCP servers and prevent errors due to user mis-configuration of DHCP servers.

Often DHCP snooping is used together with Dynamic ARP Inspection and IP Source Guard.

How DHCP Snooping Works

When enabled on a VLAN, DHCP snooping stands between untrusted ports (those connected to host ports) and trusted ports (those connected to DHCP servers). A VLAN with DHCP snooping enabled forwards DHCP request packets from clients and discards DHCP server reply packets on untrusted ports, and it forwards DHCP server reply packets on trusted ports to DHCP clients, as shown in the following figures:

Figure 46.2 DHCP Snooping at Work - on an Untrusted Port

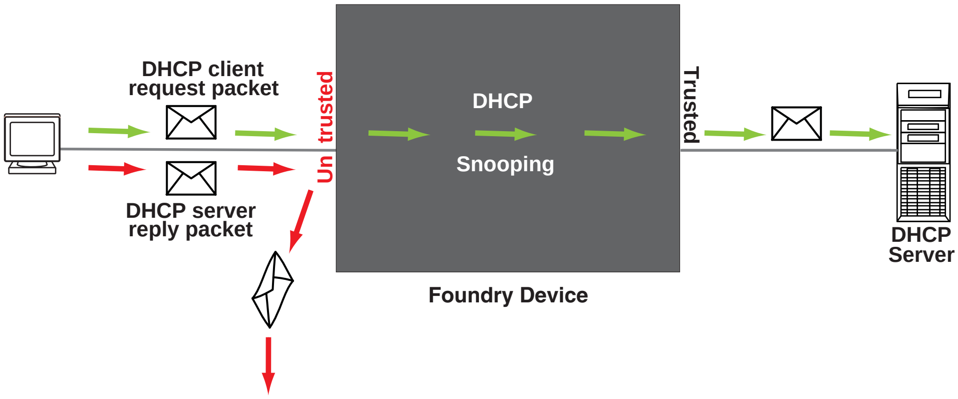
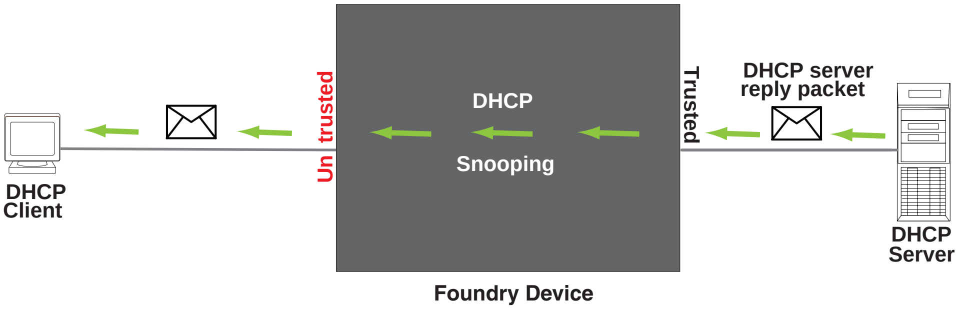


Figure 46.3 DHCP Snooping at Work - on a Trusted Port



DHCP Binding Database

When it forwards DHCP server reply packets on trusted ports, the Foundry device saves the client IP-to-MAC address binding information in the DHCP binding database. This is how the DHCP snooping binding table is populated. The information saved includes MAC address, IP address, lease time, VLAN number, and port number.

In the Foundry device, the DHCP binding database is integrated with the enhanced ARP table, which is used by Dynamic ARP Inspection. For more information, see “ARP Entries” on page 46-2.

The lease time will be refreshed when the client renews its IP address with the DHCP server; otherwise the Foundry device removes the entry when the lease time expires.

System Reboot and the Binding Database

To allow DAI and DHCP snooping to work smoothly across a system reboot, the binding database is saved to a file in the system flash memory after an update to the binding database, with a 30 second delay. The flash file is written and read only if DHCP snooping is enabled.

Configuration Notes and Feature Limitations

The following limits and restrictions apply to DHCP snooping:

- DHCP snooping is not support on trunk ports.
- A switch can have up to 256 ARP entries, therefore, DHCP entries are limited to 256. A router, however, can have 64,000 ARP entries, therefore, the router can have up to 64,000 DHCP entries, but only 1024 entries can be saved to flash on reboot.
- If you are configuring IP Source Guard on a port that belongs to more than one VLAN, you must first enable per-port-per-VLAN ACLs (ACL filtering based on VLAN membership or VE port membership). To enable this feature, enter the following command at the Global CONFIG Level of the CLI:

  ```
  FastIron(config)#enable acl-per-port-per-vlan
  FastIron(config)#write memory
  FastIron(config)#exit
  FastIron#reload
  ```

  NOTE: You must save the configuration and reload the software to place the change into effect.

- Starting in software release FSX 04.0.01, ACLs are supported on member ports of a VLAN on which DHCP snooping and Dynamic ARP Inspection (DAI) are enabled.

Configuring DHCP Snooping

Configuring DHCP snooping consists of the following steps:

1. Enable DHCP snooping on a VLAN. See “Enabling DHCP Snooping on a VLAN” on page 46-6.
2. For ports that are connected to a DHCP server, change their trust setting to trusted. See “Enabling Trust on a Port” on page 46-6.

The following shows the default settings of DHCP snooping:

Feature	Default
DHCP snooping	Disabled
Trust setting for ports	Untrusted

Enabling DHCP Snooping on a VLAN

DHCP packets for a VLAN with DHCP snooping enabled are inspected.

DHCP snooping is disabled by default. This feature must be enabled on the client and the DHCP server VLANs. To enable DHCP snooping, enter the following global command for these VLANs:

```
FastIron(config)#ip dhcp snooping vlan 2
```

The command enables DHCP snooping on VLAN 2.

Syntax: [no] ip dhcp snooping vlan <vlan-number>

The <vlan-number> variable specifies the ID of a configured client or DHCP server VLAN.

Enabling Trust on a Port

The default trust setting for a port is untrusted. To enable trust on a port connected to a DHCP server, enter commands such as the following:

```
FastIron(config)#interface ethernet 1/1
FastIron(config-if-e10000-1/1)#dhcp snooping trust
```

Port 1/1 is connected to a DHCP server. The commands change the CLI to the interface configuration level of port 1/1 and set the trust setting of port 1/1 to trusted.

Syntax: [no] dhcp snooping trust

Clearing the DHCP Binding Database

You can clear the DHCP binding database using the CLI command **clear DHCP**. You can remove all entries in the database, or remove entries for a specific IP address only.

To remove all entries from the DHCP binding database, enter the following command:

```
FastIron#clear dhcp
```

To clear entries for a specific IP address, enter a command such as the following:

```
FastIron#clear dhcp 10.10.102.4
```

Syntax: clear dhcp [<ip-addr>]

Displaying DHCP Snooping Status and Ports

To display the DHCP snooping status for a VLAN and the trusted/untrusted ports in the VLAN, enter the following command:

```
FastIron#show ip dhcp snooping vlan 2

IP DHCP snooping VLAN 2: Enabled
```

Syntax: show ip dhcp snooping [vlan <vlan-id>]

Displaying DHCP Binding Entry and Status

To display the DHCP binding entry and its current status, use the **show arp** command.

DHCP Snooping Configuration Example

The following example configures VLAN 2 and VLAN 20, and changes the CLI to the global configuration level to enable DHCP snooping on the two VLANs. The commands are as follows:

```
FastIron(config)#vlan 2
FastIron(config-vlan-2)#untagged ethe 1/3 to 1/4
FastIron(config-vlan-2)#router-interface ve 2
FastIron(config-vlan-2)#exit
FastIron(config)#ip dhcp snooping vlan 2

FastIron(config)#vlan 20
FastIron(config-vlan-20)#untagged ethe 1/1 to 1/2
FastIron(config-vlan-20)#router-interface ve 20
FastIron(config-vlan-20)#exit
FastIron(config)#ip dhcp snooping vlan 20
```

On VLAN 2, client ports 1/3 and 1/4 are untrusted by default: all client ports are untrusted. Hence, only DHCP client request packets received on ports 1/3 and 1/4 are forwarded.

On VLAN 20, ports 1/1 and 1/2 are connected to a DHCP server. DHCP server ports are set to trusted:

```
FastIron(config)#interface ethernet 1/1
FastIron(config-if-e10000-1/1)#dhcp snooping trust
FastIron(config-if-e10000-1/1)#exit
FastIron(config)#interface ethernet 1/2
FastIron(config-if-e10000-1/2)#dhcp snooping trust
FastIron(config-if-e10000-1/2)#exit
```

Hence, DHCP sever reply packets received on ports 1/1 and 1/2 are forwarded, and client IP/MAC binding information is collected.

The example also sets the DHCP server address for the local relay agent:

```
FastIron(config)#interface ve 2
FastIron(config-vif-2)#ip address 20.20.20.1/24
FastIron(config-vif-2)#ip helper-address 30.30.30.4
FastIron(config-vif-2)#interface ve 20
FastIron(config-vif-20)#ip address 30.30.30.1/24
```

IP Source Guard

Platform Support:

- FESX/FSX/FWSX devices running software release 03.0.00 and later

You can use IP Source Guard together with Dynamic ARP Inspection on untrusted ports. See “DHCP Snooping” on page 46-4 and “Dynamic ARP Inspection” on page 46-1.

Foundry’s implementation of the IP Source Guard feature supports configuration on a port, on specific VLAN members on a port (Layer 2 devices only), and on specific ports on a virtual interface (VE) (Layer 3 device only).

When IP Source Guard is first enabled, only DHCP packets are allowed and all other IP traffic is blocked. When the system learns a valid IP address, IP Source Guard then allows IP traffic. Only the traffic with valid source IP addresses are permitted. The system learns of a valid IP address from DHCP Snooping. When it learns a valid IP address, the system permits the learned source IP address.

When a new IP source entry binding on the port is created or deleted, the ACL will be recalculated and reapplied in hardware to reflect the change in IP source binding. By default, if IP Source Guard is enabled without any IP source binding on the port, an ACL that denies all IP traffic is loaded on the port.

Configuration Notes and Feature Limitations

- Foundry devices support IP Source Guard together with user ACLs (similar to ACLs for Dot1x), as long as both features are configured at the port-level or per-port-per-VLAN level. Foundry devices do not support IP Source Guard and ACLs on the same port if one is configured at the port-level and the other is configured at the per-port-per-VLAN level.
- If you are configuring IP Source Guard on a port that belongs to more than one VLAN, you must first enable per-port-per-VLAN ACLs (ACL filtering based on VLAN membership or VE port membership). To enable this feature, enter the following command at the Global CONFIG Level of the CLI:

  ```
  FastIron(config)#enable acl-per-port-per-vlan
  FastIron(config)#write memory
  FastIron(config)#exit
  FastIron#reload
  ```

  NOTE: You must save the configuration and reload the software to place the change into effect.

  NOTE: On a FastIron IPv6 device, **acl-per-port-per-vlan** is supported on virtual interfaces, but not on routing interfaces.

- The following limitations apply when configuring IP Source Guard on Layer 3 devices:
  - You cannot enable IP Source Guard on a tagged port on a Layer 3 device. To enable IP Source Guard on a tagged port, enable it on a per-VE basis.
  - You cannot enable IP Source Guard on an untagged port with VE on a Layer 3 device. To enable IP Source Guard in this configuration, enable it on a per-VE basis.
  - There are no restrictions for Layer 2, either on the port or per-VLAN.

- You cannot enable IP Source Guard on a port that has any of the following features enabled:
    - MAC address filter
    - Rate limiting
    - Trunk port
    - 802.1x with ACLs
- A port on which IP Source Guard is enabled limits the support of IP addresses, VLANs, and ACL rules per port. An IP Source Guard port supports a maximum of:
    - 64 IP addresses
    - 64 VLANs
    - 64 rules per ACL
- The number of configured ACL rules affect the rate at which hardware resources are used when IP Source Guard is enabled. Use the **show access-list hw-usage on** command to enable hardware usage for an ACL, followed by a show access-list <access-list-id> command to determine the hardware usage for an ACL. For example:

    ```
    FastIron#show access-list hw-usage on

    FastIron#show access-list 100

    Extended IP access list 100 (hw usage : 2)
    deny ip any any (hw usage : 1)
    ```

    To provide more hardware resource for IP Source Guard addresses, modify the ACL rules so that it uses less hardware resource.

Enabling IP Source Guard on a Port

You enable IP Source Guard on DHCP snooping untrusted ports. See "DHCP Snooping" on page 46-4 for how to configure DHCP and DHCP untrusted ports.

By default, IP Source Guide is disabled. To enable IP Source Guard on a DHCP untrusted port, enter the following commands:

```
FastIron(config)#interface ethernet 1/4
FastIron(config-if-e10000-1/4)#source-guard enable
```

The commands change the CLI to the interface configuration level for port 1/4 and enable IP Source Guard on the port.

Syntax: [no] source-guard enable

Defining Static IP Source Bindings

You can manually enter valid IP addresses in the binding database. To do so, enter a command such as the following:

```
FastIron(config)#ip source binding 10.10.10.1 e 2/4 vlan 4
```

Syntax: [no] ip source binding <ip-addr> ethernet [<slotnum>/]<portnum> [vlan <vlannum>]

For <ip-addr>, enter a valid IP address.

The <slotnum> parameter is required on chassis devices.

The <portnum> parameter is a valid port number.

The [vlan <vlannum>] parameter is optional. If you enter a VLAN number, the binding applies to that VLAN only. If you do not enter a VLAN number, the static binding applies to all VLANs associated with the port. Note that since static IP source bindings consume system resources, you should avoid unnecessary bindings.

Enabling IP Source Guard Per-Port-Per-VLAN

To enable IP Source Guard per-port-per VLAN, enter commands such as the following:

```
FastIron(config)#vlan 12 name vlan12
FastIron(config-vlan-12)#untag ethernet 5 to 8
FastIron(config-vlan-12)#tag ethernet 23 to 24
FastIron(config-vlan-12)#exit
FastIron(config)#int e 23
FastIron(config-if-e1000-23)#per-vlan vlan12
FastIron(config-if-e1000-23-vlan-12))#source-guard enable
```

The commands in this example configure port-based VLAN 12, and add ports e 5 – 8 as untagged ports and ports e 23 – 24 as tagged ports to the VLAN.   The last two commands enable IP Source Guard on port e 23, a member of VLAN 12.

Syntax: [no] source-guard enable

Enabling IP Source Guard on a VE

To enable IP Source Guard on a virtual interface, enter commands such as the following:

```
FastIron(config)#vlan 2
FastIron(config-vlan-2)#tag e1
Added tagged port(s) ethe 1 to port-vlan 2
FastIron(config-vlan-2)#router-int ve 2
FastIron(config-vlan-2)#int ve 2
FastIron(config-vif-2)#source-guard enable e 1
```

Syntax: [no] source-guard enable

Displaying Learned IP Addresses

To display the learned IP addresses for IP Source Guard ports, use the CLI commands **show ip source-guard** and **show ip source-guard <vlan>**.

Chapter 47

Securing SNMP Access

This chapter describes how to use the Simple Network Management Protocol (SNMP) to manage complex networks.

SNMP Overview

SNMP is a set of protocols for managing complex networks. SNMP sends messages, called protocol data units (PDUs), to different parts of a network. SNMP-compliant devices, called agents, store data about themselves in Management Information Bases (MIBs) and return this data to the SNMP requesters.

The chapter “Securing Access to Management Functions” on page 39-1 introduced a few methods used to secure SNMP access. They included the following:

- “Using ACLs to Restrict SNMP Access” on page 39-5
- “Restricting SNMP Access to a Specific IP Address” on page 39-7
- “Restricting SNMP Access to a Specific VLAN” on page 39-9
- “Disabling SNMP Access” on page 39-12

This chapter presents additional methods for securing SNMP access to Foundry devices. It contains the following sections:

- “Establishing SNMP Community Strings” on page 47-2
- “Using the User-Based Security Model” on page 47-4
- “SNMP v3 Configuration Examples” on page 47-13
- “SNMP Version 3 Traps” on page 47-8
- “Displaying SNMP Information” on page 47-11
- “SNMP v3 Configuration Examples” on page 47-13

Restricting SNMP access using ACL, VLAN, or a specific IP address constitute the first level of defense when the packet arrives at a Foundry device. The next level uses one of the following methods:

- Community string match In SNMP versions 1 and 2
- User-based model in SNMP version 3

SNMP views are incorporated in community strings and the user-based model.

Establishing SNMP Community Strings

SNMP versions 1 and 2 use community strings to restrict SNMP access. The default passwords for Web management access are the SNMP community strings configured on the device.

- The default read-only community string is “public”. To open a read-only Web management session, enter “get” and “public” for the user name and password.
- here is no default read-write community string. Thus, by default, you cannot open a read-write management session using the Web management interface. You first must configure a read-write community string using the CLI. Then you can log on using “set” as the user name and the read-write community string you configure as the password.

You can configure as many additional read-only and read-write community strings as you need. The number of strings you can configure depends on the memory on the device. There is no practical limit.

The Web management interface supports only one read-write session at a time. When a read-write session is open on the Web management interface, subsequent sessions are read-only, even if the session login is “set” with a valid read-write password.

NOTE: If you delete the startup-config file, the device automatically re-adds the default “public” read-only community string the next time you load the software.

NOTE: As an alternative to the SNMP community strings, you can secure Web management access using local user accounts or ACLs. See “Setting up Local User Accounts” on page 39-17 or “Using an ACL to Restrict Web Management Access” on page 39-5.

Encryption of SNMP Community Strings

The software automatically encrypts SNMP community strings. Users with read-only access or who do not have access to management functions in the CLI cannot display the strings. For users with read-write access, the strings are encrypted in the CLI but are shown in the clear in the Web management interface.

Encryption is enabled by default. You can disable encryption for individual strings or trap receivers if desired. See the next section for information about encryption.

Adding an SNMP Community String

When you add a community string, you can specify whether the string is encrypted or clear. By default, the string is encrypted.

To add an encrypted community string, enter commands such as the following:

```
FastIron(config)#snmp-server community private rw
FastIron(config)#write memory
```

Syntax: snmp-server community [0 | 1] <string>
ro | rw [view <viewname>] [<standard-acl-name> | <standard-acl-id>]

The <string> parameter specifies the community string name. The string can be up to 32 characters long.

The **ro | rw** parameter specifies whether the string is read-only (**ro**) or read-write (**rw**).

The **0** | **1** parameter affects encryption for display of the string in the running-config and the startup-config file. Encryption is enabled by default. When encryption is enabled, the community string is encrypted in the CLI regardless of the access level you are using. In the Web management interface, the community string is encrypted at the read-only access level but is visible at the read-write access level.

The encryption option can be omitted (the default) or can be one of the following:

- **0** – Disables encryption for the community string you specify with the command. The community string is shown as clear text in the running-config and the startup-config file. Use this option if you do not want the display of the community string to be encrypted.

- **1** – Assumes that the community string you enter is encrypted, and decrypts the value before using it.

NOTE: If you want the software to assume that the value you enter is the clear-text form, and to encrypt display of that form, do not enter **0** or **1**. Instead, omit the encryption option and allow the software to use the default behavior.

If you specify encryption option **1**, the software assumes that you are entering the encrypted form of the community string. In this case, the software decrypts the community string you enter before using the value for authentication. If you accidentally enter option **1** followed by the clear-text version of the community string, authentication will fail because the value used by the software will not match the value you intended to use.

The command in the example above adds the read-write SNMP community string "private". When you save the new community string to the startup-config file (using the **write memory** command), the software adds the following command to the file:

```
snmp-server community 1 <encrypted-string> rw
```

To add a non-encrypted community string, you must explicitly specify that you do not want the software to encrypt the string. Here is an example:

```
FastIron(config)#snmp-server community 0 private rw
FastIron(config)#write memory
```

The command in this example adds the string "private" in the clear, which means the string is displayed in the clear. When you save the new community string to the startup-config file, the software adds the following command to the file:

```
snmp-server community 0 private rw
```

The **view** <viewname> parameter is optional. It allows you to associate a view to the members of this community string. Enter up to 32 alphanumeric characters. If no view is specified, access to the full MIB is granted. The view that you want must exist before you can associate it to a community string. Here is an example of how to use the view parameter in the community string command:

```
FastIron(config)#snmp-s community myread ro view sysview
```

The command in this example associates the view "sysview" to the community string named "myread". The community string has read-only access to "sysview". For information on how to create views, see the section "SNMP v3 Configuration Examples" on page 47-13.

The <standard-acl-name> | <standard-acl-id> parameter is optional. It allows you to specify which ACL group will be used to filter incoming SNMP packets. You can enter either the ACL name or its ID. Here are some examples:

```
FastIron(config) #snmp-s community myread ro view sysview 2
FastIron(config) #snmp-s community myread ro view sysview myacl
```

The command in the first example indicates that ACL group 2 will filter incoming SNMP packets; whereas, the command in the second example uses the ACL group called "myacl" to filter incoming packets. See "Using ACLs to Restrict SNMP Access" on page 39-5 for more information.

NOTE: To make configuration changes, including changes involving SNMP community strings, you must first configure a read-write community string using the CLI. Alternatively, you must configure another authentication method and log on to the CLI using a valid password for that method.

Displaying the SNMP Community Strings

To display the configured community strings, enter the following command at any CLI level:

```
FastIron#show snmp server
Contact: Marshall
Location: Copy Center
Community(ro): public
Community(rw): private
Traps
                   Cold start: Enable
                      Link up: Enable
                    Link down: Enable
               Authentication: Enable
     Locked address violation: Enable
         Power supply failure: Enable
                  Fan failure: Enable
          Temperature warning: Enable
                 STP new root: Enable
          STP topology change: Enable
                         ospf: Enable

 Total Trap-Receiver Entries: 4
Trap-Receiver IP Address      Community
     1        207.95.6.211
     2        207.95.5.21
```

Syntax: show snmp server

NOTE: If display of the strings is encrypted, the strings are not displayed. Encryption is enabled by default.

Using the User-Based Security Model

Platform Support:

- FESX/FSX/FWSX devices running software release 03.0.00 and later
- FGS and FLS devices running software release 03.2.00 and later

SNMP version 3 (RFC 2570 through 2575) introduces a User-Based Security model (RFC 2574) for authentication and privacy services.

SNMP version 1 and version 2 use community strings to authenticate SNMP access to management modules. This method can still be used for authentication. In SNMP version 3, the User-Based Security model of SNMP can be used to secure against the following threats:

- Modification of information
- Masquerading the identity of an authorized entity
- Message stream modification
- Disclosure of information

SNMP version 3 also supports View-Based Access Control Mechanism (RFC 2575) to control access at the PDU level. It defines mechanisms for determining whether or not access to a managed object in a local MIB by a remote principal should be allowed. (See the section “SNMP v3 Configuration Examples” on page 47-13.)

Configuring Your NMS

In order to use the SNMP version 3 features:

1. Make sure that your Network Manager System (NMS) supports SNMP version 3.
2. Configure your NMS agent with the necessary users.
3. Configure the SNMP version 3 features in Foundry devices.

Configuring SNMP Version 3 on Foundry Devices

To configure SNMP version 3 on Foundry devices, do the following:

1. Enter an engine ID for the management module using the **snmp-server engineid** command if you will not use the default engine ID. See "Defining the Engine ID" on page 47-5.
2. Create views that will be assigned to SNMP user groups using the **snmp-server view** command. See "SNMP v3 Configuration Examples" on page 47-13 for details.
3. Create ACL groups that will be assigned to SNMP user groups using the **access-list** command.
4. Create user groups using the **snmp-server group** command. See "Defining an SNMP Group" on page 47-6.
5. Create user accounts and associate these accounts to user groups using the **snmp-server user** command. See "Defining an SNMP User Account" on page 47-6.

If SNMP version 3 is not configured, then community strings by default are used to authenticate access.

Defining the Engine ID

A default engine ID is generated during system start up. To determine what the default engine ID of the device is, enter the **show snmp engineid** command and find the following line.

```
Local SNMP Engine ID: 800007c70300e05290ab60
```

See the section "Displaying the Engine ID" on page 47-11 for details.

The default engine ID guarantees the uniqueness of the engine ID for SNMP version 3. If you want to change the default engine ID, enter a command such as the following:

```
FastIron(config)#snmp-server engineid local 800007c70300e05290ab60
```

Syntax: [no] snmp-server engineid local <hex-string>

The **local** parameter indicates that engine ID to be entered is the ID of this device, representing an SNMP management entity.

NOTE: Since the current implementation of SNMP version 3 does not support Notification, remote engine IDs cannot be configured at this time.

The <hex-string> variable consists of 11 octets, entered as hexadecimal values. There are two hexadecimal characters in each octet. There should be an even number of hexadecimal characters in an engine ID.

The default engine ID has a maximum of 11 octets:

- Octets 1 through 4 represent the agent's SNMP management private enterprise number as assigned by the Internet Assigned Numbers Authority (IANA). The most significant bit of Octet 1 is "1". For example, "000007c7" is the ID for Foundry Networks in hexadecimal. With Octet 1 always equal to "1", the first four octets in the default engine ID is always "800007c7" (which is 1991 in decimal).
- Octet 5 is always 03 in hexadecimal and indicates that the next set of values represent a MAC address.
- Octets 6 through 11 form the MAC address of the lowest port in the management module.

NOTE: Engine ID must be a unique number among the various SNMP engines in the management domain. Using the default engine ID ensures the uniqueness of the numbers.

Defining an SNMP Group

SNMP groups map SNMP users to SNMP views. For each SNMP group, you can configure a read view, a write view, or both. Users who are mapped to a group will use its views for access control.

To configure an SNMP user group, enter a command such as the following:

```
FastIron(config)#snmp-server group admin v3 auth read all write all
```

Syntax: [no] snmp-server group <groupname> v1 | v2 | v3 auth | noauth | priv [access <standard-acl-id>] [read <viewstring> | write <viewstring>]

NOTE: This command is not used for SNMP version 1 and SNMP version 2. In these versions, groups and group views are created internally using community strings. (See “Establishing SNMP Community Strings” on page 47-2.) When a community string is created, two groups are created, based on the community string name. One group is for SNMP version 1 packets, while the other is for SNMP version 2 packets.

The **group** <groupname> parameter defines the name of the SNMP group to be created.

The **v1**, **v2**, or **v3** parameter indicates which version of SNMP is used. In most cases, you will be using v3, since groups are automatically created in SNMP versions 1 and 2 from community strings.

The **auth** | **noauth** parameter determines whether or not authentication will be required to access the supported views. If auth is selected, then only authenticated packets are allowed to access the view specified for the user group. Selecting **noauth** means that no authentication is required to access the specified view. Selecting **priv** means that an authentication password will be required from the users.

The **access** <standard-acl-id> parameter is optional. It allows incoming SNMP packets to be filtered based on the standard ACL attached to the group.

The **read** <viewstring> | **write** <viewstring> parameter is optional. It indicates that users who belong to this group have either read or write access to the MIB.

The <viewstring> variable is the name of the view to which the SNMP group members have access. If no view is specified, then the group has no access to the MIB.

The value of <viewstring> is defined using the **snmp-server view** command. The SNMP agent comes with the "all" default view, which provides access to the entire MIB; however, it must be specified when creating the group. The "all" view also allows SNMP version 3 to be backwards compatibility with SNMP version 1 and version 2.

NOTE: If you will be using a view other than the "all" view, that view must be configured before creating the user group. See the section “SNMP v3 Configuration Examples” on page 47-13, especially for details on the include | exclude parameters.

Defining an SNMP User Account

The **snmp-server user** command does the following:

- Creates an SNMP user.
- Defines the group to which the user will be associated.
- Defines the type of authentication to be used for SNMP access by this user.

Here is an example of how to create the account:

```
FastIron(config)#snmp-s user bob admin v3 access 2 auth md5 bobmd5 priv des bobdes
```

The CLI for creating SNMP version 3 users has been updated as follows.

Syntax: [no] snmp-server user <name> <groupname> v3
[[access <standard-acl-id>] [encrypted] [auth md5 <md5-password> | sha <sha-password>]
[priv [encrypted] des <des-password>]]

The <name> parameter defines the SNMP user name or security name used to access the management module.

The <groupname> parameter identifies the SNMP group to which this user is associated or mapped. All users must be mapped to an SNMP group. Groups are defined using the **snmp-server group** command.

NOTE: The SNMP group to which the user account will be mapped should be configured before creating the user accounts; otherwise, the group will be created without any views. Also, ACL groups must be configured before configuring user accounts.

The **v3** parameter is required.

The **access** <standard-acl-id> parameter is optional. It indicates that incoming SNMP packets are filtered based on the ACL attached to the user account.

NOTE: The ACL specified in a user account overrides the ACL assigned to the group to which the user is mapped. If no ACL is entered for the user account, then the ACL configured for the group will be used to filter packets.

The **encrypted** parameter means that the MD5 or SHA password will be a digest value. MD5 has 16 octets in the digest. SHA has 20. The digest string has to be entered as a hexadecimal string. In this case, the agent need not generate any explicit digest. If the **encrypted** parameter is not used, the user is expected to enter the authentication password string for MD5 or SHA. The agent will convert the password string to a digest, as described in RFC 2574.

The **auth md5 | sha** parameter is optional. It defines the type of encryption that the user must have to be authenticated. Choose between MD5 or SHA encryption. MD5 and SHA are two authentication protocols used in SNMP version 3.

The <md5-password> and <sha-password> define the password the user must use to be authenticated. These password must have a minimum of 8 characters. If the encrypted parameter is used, then the digest has 16 octets for MD5 or 20 octets for SHA.

NOTE: Once a password string is entered, the generated configuration displays the digest (for security reasons), not the actual password.

The **priv [encrypted] des** <des-password> parameter is optional. It defines the type of encryption that will be used to encrypt the privacy password. If the "encryption" keyword is used, enter a 16-octet DES key in hexadecimal format for the des-password. If the "encryption" keyword is not used enter a password string. The agent will generate a suitable 16-octet DES key from the password string.

Currently, DES is the only encryption type supported for priv password.

Defining SNMP Views

SNMP views are named groups of MIB objects that can be associated with user accounts to allow limited access for viewing and modification of SNMP statistics and system configuration. SNMP views can also be used with other commands that take SNMP views as an argument. SNMP views reference MIB objects using object names, numbers, wildcards, or a combination of the three. The numbers represent the hierarchical location of the object in the MIB tree. You can reference individual objects in the MIB tree or a subset of objects from the MIB tree.

To configure the number of SNMP views available on the Foundry device:

```
FastIron(config)#system-max view 15
```

Syntax: system-max view <number-of-views>

This command specifies the maximum number of SNMPv2 and v3 views that can be configured on a device. The number of views can be from 10 – 65536. The default is 10 views.

To add an SNMP view, enter one of the following commands:

```
FastIron(config)#snmp-server view Maynes system included
FastIron(config)#snmp-server view Maynes system.2 excluded
FastIron(config)#snmp-server view Maynes 2.3.*.6 included
FastIron(config)#write mem
```

NOTE: The **snmp-server view** command supports the MIB objects as defined in RFC 1445.

Syntax: [no] snmp-server view <name> <mib_tree> included | excluded

The <name> parameter can be any alphanumeric name you choose to identify the view. The names cannot contain spaces.

The <mib_tree> parameter is the name of the MIB object or family. MIB objects and MIB sub-trees can be identified by a name or by the numbers called Object Identifiers (OIDs) that represent the position of the object or sub-tree in the MIB hierarchy. You can use a wildcard (*) in the numbers to specify a sub-tree family.

The **included** | **excluded** parameter specifies whether the MIB objects identified by the <mib_family> parameter are included in the view or excluded from the view.

NOTE: All MIB objects are automatically excluded from any view unless they are explicitly included; therefore, when creating views using the **snmp-server view** command, indicate which portion of the MIB you want users to access.

For example, you may want to assign the view called “admin” a community string or user group. The “admin” view will allow access to the Foundry MIBs objects that begin with the 1.3.6.1.4.1.1991 object identifier. Enter the following command:

```
FastIron(config)#snmp-server view admin 1.3.6.1.4.1.1991 included
```

You can exclude portions of the MIB within an inclusion scope. For example, if you want to exclude the snAgentSys objects, which begin with 1.3.6.1.4.1.1991.1.1.2 object identifier from the admin view, enter a second command such as the following:

```
FastIron(config)#snmp-server view admin 1.3.6.1.4.1.1991.1.1.2 excluded
```

Note that the exclusion is within the scope of the inclusion.

To delete a view, use the no parameter before the command.

SNMP Version 3 Traps

Platform Support:

- FESX/FSX/FWSX devices running software release 03.0.00 and later
- FGS and FLS devices running software release 03.2.00 and later

FastIron devices support SNMP notifications in SMIv2 format. This allows notifications to be encrypted and sent to the target hosts in a secure manner.

Defining an SNMP Group and Specifying Which View is Notified of Traps

The SNMP group command allows configuration of a viewname for notification purpose, similar to the read and write view. The default viewname is "all", which allows access to the entire MIB.

To configure an SNMP user group, first configure SNMP v3 views using the **snmp-server view** command. See “SNMP v3 Configuration Examples” on page 47-13. Then enter a command such as the following:

```
FastIron(config)#snmp-server group admin v3 auth read all write all
notify all
```

Syntax: [no] snmp-server group <groupname>
v1 | v2 | v3

auth | noauth | priv
[access <standard-acl-id>] [read <viewstring> | write <viewstring> | notify <viewstring>]

The **group** <groupname> parameter defines the name of the SNMP group to be created.

The **v1**, **v2**, or **v3** parameter indicates which version of SNMP to use. In most cases, you will use v3, since groups are automatically created in SNMP versions 1 and 2 from community strings.

The **auth** | **noauth** parameter determines whether or not authentication will be required to access the supported views. If auth is selected, then only authenticated packets are allowed to access the view specified for the user group. Selecting **noauth** means that no authentication is required to access the specified view. Selecting **priv** means that an authentication password will be required from the users.

The **access** <standard-acl-id> parameter is optional. It allows incoming SNMP packets to be filtered based on the standard ACL attached to the group.

The **read** <viewstring> | **write** <viewstring> parameter is optional. It indicates that users who belong to this group have either read or write access to the MIB.

The **notify** view allows administrators to restrict the scope of varbind objects that will be part of the notification. All of the varbinds need to be in the included view for the notification to be created.

The <viewstring> variable is the name of the view to which the SNMP group members have access. If no view is specified, then the group has no access to the MIB.

Defining the UDP Port for SNMP v3 Traps

Platform Support:

- FGS and FLS devices running software release 03.2.00 and later

The SNMP host command enhancements allow configuration of notifications in SMIv2 format, with or without encryption, in addition to the previously supported SMIv1 trap format.

You can define a port that receives the SNMP v3 traps by entering a command such as the following:

```
FastIron(config)#snmp-server host 192.168.4.11 version v3 auth security-name port 4/1
[no] snmp-server host <ip-addr> | <ipv6-addr> version [ v1 | v2c <community-string> | v3 auth
| noauth | priv <security-name>]  [port <trap-UDP-port-number>]
```

The <ip-addr> parameter specifies the IP address of the host that will receive the trap.

For **version**, indicate one of the following:

For SNMP version 1, enter **v1** and the name of the community string (<community-string>). This string is encrypted within the system.

NOTE: The options "v2c" and "v3" are new in software release 03.0.00. If the configured version is v2c, then the notification is sent out in SMIv2 format, using the community string, but in cleartext mode. To send the SMIv2 notification in SNMPv3 packet format, configure v3 with auth and/or privacy parameters by specifying a security name. The actual authorization and privacy values are obtained from the security name.

For SNMP version 2c, enter **v2** and the name of the community string. This string is encrypted within the system.

For SNMP version 3, enter one of the following depending on the authorization required for the host:

- **v3 auth** <security-name>: Allow only authenticated packets.
- **v3 no auth** <security-name>: Allow all packets.
- **v3 priv** <security-name>: A password is required

For **port** <trap-UDP-port-number>, specify the UDP port number on the host that will receive the trap.

Trap MIB Changes

To support the SNMP V3 trap feature, the Foundry Enterprise Trap MIB was rewritten in SMIv2 format, as follows:

- The M-IB name was changed from FOUNDRY-SN-TRAP-MIB to FOUNDRY-SN-NOTIFICATION-MIB
- Individual notifications were changed to NOTIFICATION-TYPE instead of TRAP-TYPE.
- As per the SMIv2 format, each notification has an OID associated with it. The root node of the notification is snTraps (OID: enterprise.foundry.0). For example, OID for snTrapRunningConfigChanged is {snTraps.73}. Earlier, each trap had a trap ID associated with it, as per the SMIv1 format.

Backward compatibility with SMIv1 trap format

The Foundry device will continue to support creation of traps in SMIv1 format, as before. To allow the device to send notifications in SMIv2 format, configure the device as described above. The default mode is still the original SMIv1 format.

Restricting SNMP Access to an IPv6 Node

You can restrict SNMP access so that the device (including IronView Network Manager) can only be accessed by the IPv6 host's address that you specify. To do so, enter a command such as the following:

```
FastIron(config)# snmp-client ipv6 2001:efff:89::23
```

Syntax: snmp-client ipv6 <ipv6-address>

The <ipv6-address> must be in hexadecimal format using 16-bit values between colons as documented in RFC 2373.

Specifying an IPv6 Host as an SNMP Trap Receiver

You can specify an IPv6 host as a trap receiver to ensure that all SNMP traps sent by the device will go to the same SNMP trap receiver or set of receivers, typically one or more host devices on the network. To do so, enter a command such as the following:

```
FastIron(config)# snmp-server host ipv6 2001:efff:89::13
```

Syntax: snmp-server host ipv6 <ipv6-address>

The <ipv6-address> must be in hexadecimal format using 16-bit values between colons as documented in RFC 2373.

Possible values: N/A

Default value: N/A

SNMP3 over IPv6

Platform Support:

- FESX/FSX/FWSX devices running software release 02.4.00 and later
- FGS and FLS devices running software release 04.0.00 and later

Beginning with the releases shown above, Foundry FastIron devices support IPv6 for SNMP version 3.

Restricting SNMP Access to an IPv6 Node

You can restrict SNMP access so that the Foundry device (including IronView Network Manager) can only be accessed by the IPv6 host's address that you specify. To do so, enter a command such as the following:

```
FastIron(config)#snmp-client ipv6 2001:efff:89::23
```

Syntax: snmp-client ipv6 <ipv6-address>

The <ipv6-address> must be in hexadecimal format using 16-bit values between colons as documented in RFC 2373.

Specifying an IPv6 Host as an SNMP Trap Receiver

You can specify an IPv6 host as a trap receiver to ensure that all SNMP traps sent by the Foundry device will go to the same SNMP trap receiver or set of receivers, typically one or more host devices on the network. To do so, enter a command such as the following:

```
FES Switch(config)#snmp-server host ipv6 2001:efff:89::13
```

Syntax: snmp-server host ipv6 <ipv6-address>

The <ipv6-address> must be in hexadecimal format using 16-bit values between colons as documented in RFC 2373.

Possible values: N/A

Default value: N/A

Viewing IPv6 SNMP Server Addresses

Platform Support:

- FESX/FSX/FWSX devices running software release 02.4.00 and later
- FGS and FLS devices running software release 04.0.00 and later

Starting with the releases shown above, many of the existing **show** commands now display IPv6 addresses for IPv6 SNMP servers. The following example shows output for the **show snmp server** command.

```
FastIron#show snmp server
      Contact:
     Location:
Community(ro): .....
Traps
              Warm/Cold start: Enable
                      Link up: Enable
                    Link down: Enable
               Authentication: Enable
     Locked address violation: Enable
         Power supply failure: Enable
                  Fan failure: Enable
          Temperature warning: Enable
                 STP new root: Enable
          STP topology change: Enable
                         vsrp: Enable

  Total Trap-Receiver Entries: 4

Trap-Receiver IP-Address                Port-Number Community
     1        192.147.201.100               162     .....
     2        4000::200                     162     .....
      3        192.147.202.100               162     .....
      4        3000::200                     162     .....
```

Displaying SNMP Information

This section lists the commands for viewing SNMP-related information.

Displaying the Engine ID

To display the engine ID of a management module, enter a command such as the following:

```
FastIron#show snmp engineid
Local SNMP Engine ID: 800007c70300e05290ab60
```

```
Engine Boots: 3
Engine time: 5
```

Syntax: show snmp engineid

The engine ID identifies the source or destination of the packet.

The engine boots represents the number of times that the SNMP engine reinitialized itself with the same engine ID. If the engineID is modified, the boot count is reset to 0.

The engine time represents the current time with the SNMP agent.

Displaying SNMP Groups

To display the definition of an SNMP group, enter a command such as the following:

```
FastIron#show snmp group
groupname = exceptifgrp
security model = v3
security level = authNoPriv
ACL id = 2
readview = exceptif
writeview = <none>
```

Syntax: show snmp group

The value for security level can be one of the following:

Security Level	Authentication
<none>	If the security model shows v1 or v2, then security level is blank. User names are not used to authenticate users; community strings are used instead.
noauthNoPriv	Displays if the security model shows v3 and user authentication is by user name only.
authNoPriv	Displays if the security model shows v3 and user authentication is by user name and the MD5 or SHA algorithm.

Displaying User Information

To display the definition of an SNMP user account, enter a command such as the following:

```
FastIron#show snmp user

username = bob
acl id = 2
group = admin
security model = v3
group acl id = 0
authtype = md5
authkey = 3aca18d90b8d172760e2dd2e8f59b7fe
privtype = des,  privkey = 1088359afb3701730173a6332d406eec
engine ID= 800007c70300e052ab0000
```

Syntax: show snmp user

Interpreting Varbinds in Report Packets

If an SNMP version 3 request packet is to be rejected by an SNMP agent, the agent sends a report packet that contains one or more varbinds. The varbinds contain additional information, showing the cause of failures. An

SNMP manager application decodes the description from the varbind. The following table presents a list of varbinds supported by the SNMP agent.

Varbind Object Identifier	**Description**
1. 3. 6. 1. 6. 3. 11. 2. 1. 3. 0	Unknown packet data unit.
1. 3. 6. 1. 6. 3. 12. 1. 5. 0	The value of the varbind shows the engine ID that needs to be used in the snmp-server engineid command
1. 3. 6. 1. 6. 3. 15. 1. 1. 1. 0	Unsupported security level.
1. 3. 6. 1. 6. 3. 15. 1. 1. 2. 0	Not in time packet.
1. 3. 6. 1. 6. 3. 15. 1. 1. 3. 0	Unknown user name. This varbind may also be generated: • If the configured ACL for this user filters out this packet. • If the group associated with the user is unknown.
1. 3. 6. 1. 6. 3. 15. 1. 1. 4. 0	Unknown engine ID. The value of this varbind would be the correct authoritative engineID that should be used.
1. 3. 6. 1. 6. 3. 15. 1. 1. 5. 0	Wrong digest.
1. 3. 6. 1. 6. 3. 15. 1. 1. 6. 0	Decryption error.

SNMP v3 Configuration Examples

The following sections present examples of how to configure SNMP v3.

Simple SNMP v3 Configuration

```
FastIron(config)#snmp-s group admingrp v3 priv read all write all notify all
FastIron(config)#snmp-s user adminuser admingrp v3 auth md5 <auth password> priv
<privacy password>
FastIron(config)#snmp-s host <dest-ip> version v3 privacy adminuser
```

More Detailed SNMP v3 Configuration

```
FastIron(config)#snmp-server view internet internet included
FastIron(config)#snmp-server view system system included
FastIron(config)#snmp-server community ..... ro
FastIron(config)#snmp-server community ..... rw
FastIron(config)#snmp-server contact isc-operations
FastIron(config)#snmp-server location sdh-pillbox
FastIron(config)#snmp-server host 128.91.255.32 .....
FastIron(config)#snmp-server group ops v3 priv read internet write system
FastIron(config)#snmp-server group admin v3 priv read internet write internet
FastIron(config)#snmp-server group restricted v3 priv read internet
FastIron(config)#snmp-server user ops ops v3 encrypted auth md5
ab8e9cd6d46e7a270b8c9549d92a069 priv encrypted des 0e1b153303b6188089411447dbc32de
FastIron(config)#snmp-server user admin admin v3 encrypted auth md5
0d8a2123f91bfbd8695fef16a6f4207b priv encrypted des
18e0cf359fce4fcd60df19c2b6515448
FastIron(config)#snmp-server user restricted restricted v3 encrypted auth md5
261fd8f56a3ad51c8bcec1e4609f54dc priv encrypted des
d32e66152f89de9b2e0cb17a65595f43
```

Appendix A
Using Syslog

This appendix describes how to display Syslog messages and how to configure the Syslog facility, and lists the Syslog messages that Foundry devices can display during standard operation.

NOTE: This appendix does not list Syslog messages that can be displayed when a debug option is enabled.

Overview

A Foundry device's software can write syslog messages to provide information at the following severity levels:

- Emergencies
- Alerts
- Critical
- Errors
- Warnings
- Notifications
- Informational
- Debugging

The device writes the messages to a local buffer. The buffer can hold up to 1000 entries.

You also can specify the IP address or host name of up to six Syslog servers. When you specify a Syslog server, the Foundry device writes the messages both to the system log and to the Syslog server.

Using a Syslog server ensures that the messages remain available even after a system reload. The Foundry device's local Syslog buffer is cleared during a system reload or reboot, but the Syslog messages sent to the Syslog server remain on the server.

NOTE: Starting in software release 02.5.00, you can enable the Foundry device to retain Syslog messages after a soft reboot (**reload** command). See "Retaining Syslog Messages After a Soft Reboot" on page A-9.

The Syslog service on a Syslog server receives logging messages from applications on the local host or from devices such as a Layer 2 Switch or Layer 3 Switch. Syslog adds a time stamp to each received message and directs messages to a log file. Most Unix workstations come with Syslog configured. Some third party vendor products also provide Syslog running on NT.

Syslog uses UDP port 514 and each Syslog message thus is sent with destination port 514. Each Syslog message is one line with Syslog message format. The message is embedded in the text portion of the Syslog format. There are several subfields in the format. Keywords are used to identify each subfield, and commas are delimiters. The subfield order is insensitive except that the text subfield should be the last field in the message. All the subfields are optional.

Displaying Syslog Messages

To display the Syslog messages in the device's local buffer, enter the following command at any level of the CLI:

```
FastIron>#show logging

Syslog logging: enabled (0 messages dropped, 0 flushes, 0 overruns)
    Buffer logging: level ACDMEINW, 3 messages logged
    level code: A=alert C=critical D=debugging M=emergency E=error
                I=informational N=notification W=warning

Static Log Buffer:
Dec 15 19:04:14:A:Fan 1, fan on right connector, failed

Dynamic Log Buffer (50 entries):
Dec 15 18:46:17:I:Interface ethernet 4, state up
Dec 15 18:45:21:I:Bridge topology change, vlan 4095, interface 4, changed
state to forwarding
Dec 15 18:45:15:I:Warm start
```

For information about the Syslog configuration information, time stamps, and dynamic and static buffers, see "Displaying the Syslog Configuration" on page A-4.

Enabling Real-Time Display of Syslog Messages

By default, to view Syslog messages generated by a Foundry device, you need to display the Syslog buffer or the log on a Syslog server used by the Foundry device.

You can enable real-time display of Syslog messages on the management console. When you enable this feature, the software displays a Syslog message on the management console when the message is generated. However, to enable display of real-time Syslog messages in Telnet or SSH sessions, you also must enable display within the individual sessions.

To enable real-time display of Syslog messages, enter the following command at the global CONFIG level of the CLI:

```
FastIron(config)#logging console
```

Syntax: [no] logging console

This command enables the real-time display of Syslog messages on the serial console. You can enter this command from the serial console or a Telnet or SSH session.

Enabling Real-Time Display for a Telnet or SSH Session

To also enable the real-time display for a Telnet or SSH session, enter the following command from the Privileged EXEC level of the session:

```
telnet@FastIron#terminal monitor
Syslog trace was turned ON
```

Syntax: terminal monitor

Notice that the CLI displays a message to indicate the status change for the feature. To disable the feature in the management session, enter the **terminal monitor** command again. The command toggles the feature on and off.

```
telnet@FastIron#terminal monitor
```

```
Syslog trace was turned OFF
```

Here is an example of how the Syslog messages are displayed:

```
telnet@FastIron#terminal monitor
Syslog trace was turned ON
SYSLOG: <9>FastIron, Power supply 2, power supply on left connector, failed

SYSLOG: <14>FastIron, Interface ethernet 6, state down

SYSLOG: <14>FastIron, Interface ethernet 2, state up
```

Show Log on All Terminals

Platform Support:

- FGS and FLS devices running software release 04.1.00 and later
- FESX/FSX/FWSX devices running software release 04.1.00 and later – L2, BL3, L3

In software versions prior to 04.1.00, log messages could only be sent to a single terminal logged-on to a FastIron switch. Software version 04.1.00 removes that limitation and permits any terminal logged on to a FastIron switch to receive real-time Syslog messages when the **terminal monitor** command is issued.

Configuring the Syslog Service

The procedures in this section describe how to perform the following Syslog configuration tasks:

- Specify a Syslog server. You can configure the Foundry device to use up to six Syslog servers. (Use of a Syslog server is optional. The system can hold up to 100 Syslog messages in an internal buffer.)
- Change the level of messages the system logs.
- Change the number of messages the local Syslog buffer can hold.
- Display the Syslog configuration.
- Clear the local Syslog buffer.

Logging is enabled by default, with the following settings:

- Messages of all severity levels (Emergencies – Debugging) are logged.
- By default, up to 50 messages are retained in the local Syslog buffer. This can be changed.
- No Syslog server is specified.

Displaying the Syslog Configuration

To display the Syslog parameters currently in effect on a Foundry device, enter the following command from any level of the CLI:

```
FastIron>#show logging

Syslog logging: enabled (0 messages dropped, 0 flushes, 0 overruns)
    Buffer logging: level ACDMEINW, 3 messages logged
    level code: A=alert C=critical D=debugging M=emergency E=error
                I=informational N=notification W=warning

Static Log Buffer:
Dec 15 19:04:14:A:Fan 1, fan on right connector, failed

Dynamic Log Buffer (50 entries):
Dec 15 18:46:17:I:Interface ethernet 1/4, state up
Dec 15 18:45:21:I:Bridge topology change, vlan 4095, interface 4, changed
state to forwarding
Dec 15 18:45:15:I:Warm start
```

Syntax: show logging

The Syslog display shows the following configuration information, in the rows above the log entries themselves.

Table 47.1: CLI Display of Syslog Buffer Configuration

This Field...	Displays...
Syslog logging	The state (enabled or disabled) of the Syslog buffer.
messages dropped	The number of Syslog messages dropped due to user-configured filters. By default, the software logs messages for all Syslog levels. You can disable individual Syslog levels, in which case the software filters out messages at those levels. See “Disabling Logging of a Message Level” on page A-7. Each time the software filters out a Syslog message, this counter is incremented.
flushes	The number of times the Syslog buffer has been cleared by the **clear logging** command or equivalent Web management interface option. See “Clearing the Syslog Messages from the Local Buffer” on page A-10.
overruns	The number of times the dynamic log buffer has filled up and been cleared to hold new entries. For example, if the buffer is set for 100 entries, the 101st entry causes an overrun. After that, the 201st entry causes a second overrun.
level	The message levels that are enabled. Each letter represents a message type and is identified by the key (level code) below the value. If you disable logging of a message level, the code for that level is not listed.
messages logged	The total number of messages that have been logged since the software was loaded.
level code	The message levels represented by the one-letter codes.

Static and Dynamic Buffers

The software provides two buffers:

- Static – logs power supply failures, fan failures, and temperature warning or shutdown messages
- Dynamic – logs all other message types

In the static log, new messages replace older ones, so only the most recent message is displayed. For example, only the most recent temperature warning message will be present in the log. If multiple temperature warning messages are sent to the log, the latest one replaces the previous one. The static buffer is not configurable.

The message types that appear in the static buffer do not appear in the dynamic buffer. The dynamic buffer contains up to the maximum number of messages configured for the buffer (50 by default), then begins removing the oldest messages (at the bottom of the log) to make room for new ones.

The static and dynamic buffers are both displayed when you display the log.

```
FastIron#show logging

Syslog logging: enabled (0 messages dropped, 0 flushes, 0 overruns)
    Buffer logging: level ACDMEINW, 3 messages logged
    level code: A=alert C=critical D=debugging M=emergency E=error
          I=informational N=notification W=warning

Static Log Buffer:
Dec 15 19:04:14:A:Fan 1, fan on right connector, failed
Dec 15 19:00:14:A:Fan 2, fan on left connector, failed


Dynamic Log Buffer (50 entries):
Dec 15 18:46:17:I:Interface ethernet 4, state up
Dec 15 18:45:21:I:Bridge topology change, vlan 4095, interface 4, changed
state to forwarding
Dec 15 18:45:15:I:Warm start
```

Notice that the static buffer contains two separate messages for fan failures. Each message of each type has its own buffer. Thus, if you replace fan 1 but for some reason that fan also fails, the software replaces the first message about the failure of fan 1 with the newer message. The software does not overwrite the message for fan 2, unless the software sends a newer message for fan 2.

When you clear log entries, you can selectively clear the static or dynamic buffer, or you can clear both. For example, to clear only the dynamic buffer, enter the following command at the Privileged EXEC level:

```
FastIron#clear logging dynamic-buffer
```

Syntax: clear logging [dynamic-buffer | static-buffer]

You can specify **dynamic-buffer** to clear the dynamic buffer or **static-buffer** to clear the static buffer. If you do not specify a buffer, both buffers are cleared.

Time Stamps

The contents of the time stamp differ depending on whether you have set the time and date on the onboard system clock.

- If you have set the time and date on the onboard system clock, the date and time are shown in the following format:

  mm dd hh:*mm*:*ss*

  where:

  - *mm* – abbreviation for the name of the month
  - *dd* – day

  - *hh* – hours
  - *mm* – minutes
  - *ss* – seconds

  For example, “Oct 15 17:38:03” means October 15 at 5:38 PM and 3 seconds.

- If you have not set the time and date on the onboard system clock, the time stamp shows the amount of time that has passed since the device was booted, in the following format:

  <*num*>d<*num*>h<*num*>m<*num*>s

  where:

  - <*num*>d – day
  - <*num*>h – hours
  - <*num*>m – minutes
  - <*num*>s – seconds

  For example, “188d1h01m00s” means the device had been running for 188 days, 11 hours, one minute, and zero seconds when the Syslog entry with this time stamp was generated.

Example of Syslog Messages on a Device Whose Onboard Clock Is Set

The example shows the format of messages on a device whose onboard system clock has been set. Each time stamp shows the month, the day, and the time of the system clock when the message was generated. For example, the system time when the most recent message (the one at the top) was generated was October 15 at 5:38 PM and 3 seconds.

```
FastIron#show logging
Syslog logging: enabled (0 messages dropped, 0 flushes, 0 overruns)
    Buffer logging: level ACDMEINW, 38 messages logged
    level code: A=alert C=critical D=debugging M=emergency E=error
                I=informational N=notification W=warning

Static Log Buffer:
Dec 15 19:04:14:A:Fan 1, fan on right connector, failed
Dec 15 19:00:14:A:Fan 2, fan on left connector, failed

Dynamic Log Buffer (50 entries):
Oct 15 17:38:03:warning:list 101 denied tcp 209.157.22.191(0)(Ethernet 18
0010.5a1f.77ed) -> 198.99.4.69(http), 1 event(s)

Oct 15 07:03:30:warning:list 101 denied tcp 209.157.22.26(0)(Ethernet 18
0010.5a1f.77ed) -> 198.99.4.69(http), 1 event(s)

Oct 15 06:58:30:warning:list 101 denied tcp 209.157.22.198(0)(Ethernet 18
0010.5a1f.77ed) -> 198.99.4.69(http), 1 event(s)
```

Example of Syslog Messages on a Device Whose Onboard Clock Is Not Set

The example shows the format of messages on a device whose onboard system clock is not set. Each time stamp shows the amount of time the device had been running when the message was generated. For example, the most recent message, at the top of the list of messages, was generated when the device had been running for 21 days, seven hours, two minutes, and 40 seconds.

```
FastIron#show logging

Syslog logging: enabled (0 messages dropped, 0 flushes, 0 overruns)
    Buffer logging: level ACDMEINW, 38 messages logged
    level code: A=alert C=critical D=debugging M=emergency E=error
                I=informational N=notification W=warning

Static Log Buffer:

Dynamic Log Buffer (50 entries):
21d07h02m40s:warning:list 101 denied tcp 209.157.22.191(0)(Ethernet 4/18
0010.5a1f.77ed) -> 198.99.4.69(http), 1 event(s)

19d07h03m30s:warning:list 101 denied tcp 209.157.22.26(0)(Ethernet 4/18
0010.5a1f.77ed) -> 198.99.4.69(http), 1 event(s)

17d06h58m30s:warning:list 101 denied tcp 209.157.22.198(0)(Ethernet 4/18
0010.5a1f.77ed) -> 198.99.4.69(http), 1 event(s)
```

Disabling or Re-Enabling Syslog

Syslog is enabled by default. To disable it, enter the following command at the global CONFIG level:

```
FastIron(config)#no logging on
```

Syntax: [no] logging on [<udp-port>]

The <udp-port> parameter specifies the application port used for the Syslog facility. The default is 514.

To re-enable logging, enter the following command:

```
FastIron(config)#logging on
```

This command enables local Syslog logging with the following defaults:

- Messages of all severity levels (Emergencies – Debugging) are logged.
- Up to 50 messages are retained in the local Syslog buffer.
- No Syslog server is specified.

Specifying a Syslog Server

To specify a Syslog server, enter a command such as the following:

```
FastIron(config)#logging host 10.0.0.99
```

Syntax: logging host <ip-addr> | <server-name>

Specifying an Additional Syslog Server

To specify an additional Syslog server, enter the **logging host** <ip-addr> command again, as in the following example. You can specify up to six Syslog servers.

```
FastIron(config)#logging host 10.0.0.99
```

Syntax: logging host <ip-addr> | <server-name>

Disabling Logging of a Message Level

To change the message level, disable logging of specific message levels. You must disable the message levels on an individual basis.

For example, to disable logging of debugging and informational messages, enter the following commands:

```
FastIron(config)#no logging buffered debugging
```

```
FastIron(config)#no logging buffered informational
```

Syntax: [no] logging buffered <level> | <num-entries>

The <level> parameter can have one of the following values:

- alerts
- critical
- debugging
- emergencies
- errors
- informational
- notifications
- warnings

The commands in the example above change the log level to notification messages or higher. The software will not log informational or debugging messages. The changed message level also applies to the Syslog servers.

Changing the Number of Entries the Local Buffer Can Hold

You also can use the **logging buffered** command to change the number of entries the local Syslog buffer can store. For example:

```
FastIron(config)#logging buffered 100
FastIron(config)#write mem
FastIron(config)#exit
FastIron#reload
```

Syntax: logging buffered <1 – 100>

The default number of messages is 50. The value can be from 1 – 100 on Layer 2 Switches and Layer 3 Switches.

Configuration Notes

- In FESX/FSX/FWSX software releases 02.5.00 and later, you must save the configuration and reload the software to place the change into effect. In releases prior to 02.5.00, the change takes effect immediately and does not require you to reload the software.
- If you decrease the size of the buffer, the software clears the buffer before placing the change into effect.
- If your device is running software release 02.4.00 and you increase the size of the Syslog buffer, the software will clear some of the older locally buffered Syslog messages.

Changing the Log Facility

The Syslog daemon on the Syslog server uses a facility to determine where to log the messages from the Foundry device. The default facility for messages the Foundry device sends to the Syslog server is “user”. You can change the facility using the following command.

NOTE: You can specify only one facility. If you configure the Foundry device to use two Syslog servers, the device uses the same facility on both servers.

```
FastIron(config)#logging facility local0
```

Syntax: logging facility <facility-name>

The <facility-name> can be one of the following:

- kern – kernel messages
- user – random user-level messages
- mail – mail system
- daemon – system daemons
- auth – security/authorization messages
- syslog – messages generated internally by Syslog
- lpr – line printer subsystem
- news – netnews subsystem
- uucp – uucp subsystem
- sys9 – cron/at subsystem
- sys10 – reserved for system use
- sys11 – reserved for system use
- sys12 – reserved for system use
- sys13 – reserved for system use
- sys14 – reserved for system use
- cron – cron/at subsystem
- local0 – reserved for local use
- local1 – reserved for local use
- local2 – reserved for local use
- local3 – reserved for local use
- local4 – reserved for local use
- local5 – reserved for local use
- local6 – reserved for local use
- local7 – reserved for local use

Retaining Syslog Messages After a Soft Reboot

Platform Support:

- FESX/FSX/FWSX devices running software release 02.5.00 and later

You can configure the Foundry device to save the System log (Syslog) after a soft reboot (**reload** command). In releases prior to 02.5.00, the device clears the messages from the System log when the **reload** command is issued.

Configuration Considerations

- If the number of entries that the Syslog buffer can hold was set to a lower value using the CLI command **logging buffered**, the System log will be cleared after a soft reboot, even when the **logging persistence** feature is in effect. To prevent the device from clearing the System log, either leave the number of entries allowed in the Syslog buffer unchanged, or increase the number of entries allowed in the Syslog buffer.
- This feature does not save Syslog messages after a hard reboot. When the Foundry device is power-cycled, the Syslog messages are cleared.
- This feature is not supported in the FGS.
- If **logging persistence** is enabled and you load a new software image on the device, you must first clear the log file if you want to reload the device. (See “Clearing the Syslog Messages from the Local Buffer” on

page A-10.)

Configuration Syntax

To configure the device to save the System log messages after a soft reboot, enter the following command:

```
FastIron(config)#logging persistence
```

Syntax: [no] logging persistence

Enter **no logging persistence** to disable this feature after it has been enabled.

Clearing the Syslog Messages from the Local Buffer

To clear the Syslog messages stored in the Foundry device's local buffer, enter the following command:

```
FastIron#clear logging
```

Syntax: clear logging

Displaying TCP/UDP Port Numbers in Syslog Messages

Platform Support:

- FGS and FLS devices running software release 04.0.00 and later
- FESX/FSX/FWSX devices running software release 02.5.00 and later

The command **ip show-service-number-in-log** allows you to change the display of TCP/UDP application information from the TCP/UDP well-known port name to the TCP/UDP port number. For example, when this command is in effect, the Foundry device will display **http** (the well-known port name) instead of **80** (the port number) in the output of **show** commands, and other commands that contain application port information. By default, Foundry devices display TCP/UDP application information in named notation.

To display TCP/UDP port number instead of their names, enter the following command:

```
FastIron(config)#ip show-service-number-in-log
```

Syntax: [no] ip show-service-number-in-log

Syslog Messages

Table 47.2 lists all of the Syslog messages. Note that some of the messages apply only to Layer 3 Switches. The messages are listed by message level, in the following order, then by message type:

- Emergencies (none)
- Alerts
- Critical
- Errors
- Warnings
- Notifications
- Informational
- Debugging

Table 47.2: Foundry Syslog Messages

Message Level	Message	Explanation
Alert	<num-modules> modules and 1 power supply, need more power supply!!	Indicates that the chassis needs more power supplies to run the modules in the chassis. The <num-modules> parameter indicates the number of modules in the chassis.
Alert	Fan <num>, <location>, failed	A fan has failed. The <num> is the fan number. The <location> describes where the failed fan is in the chassis.
Alert	ISIS MEMORY USE EXCEEDED	IS-IS is requesting more memory than is available.
Alert	MAC Authentication failed for <mac-address> on <portnum>	RADIUS authentication was successful for the specified <mac-address> on the specified <portnum>; however, the VLAN returned in the RADIUS Access-Accept message did not refer to a valid VLAN or VLAN ID on the Foundry device. This is treated as an authentication failure.
Alert	MAC Authentication failed for <mac-address> on <portnum> (Invalid User)	RADIUS authentication failed for the specified <mac-address> on the specified <portnum> because the MAC address sent to the RADIUS server was not found in the RADIUS server's users database.
Alert	MAC Authentication failed for <mac-address> on <portnum> (No VLAN Info received from RADIUS server)	RADIUS authentication was successful for the specified <mac-address> on the specified <portnum>; however, dynamic VLAN assignment was enabled for the port, but the RADIUS Access-Accept message did not include VLAN information. This is treated as an authentication failure.
Alert	MAC Authentication failed for <mac-address> on <portnum> (Port is already in another radius given vlan)	RADIUS authentication was successful for the specified <mac-address> on the specified <portnum>; however, the RADIUS Access-Accept message specified a VLAN ID, although the port had previously been moved to a different RADIUS-assigned VLAN. This is treated as an authentication failure.
Alert	MAC Authentication failed for <mac-address> on <portnum> (RADIUS given vlan does not exist)	RADIUS authentication was successful for the specified <mac-address> on the specified <portnum>; however, the RADIUS Access-Accept message specified a VLAN that does not exist in the Foundry device's configuration. This is treated as an authentication failure.

Table 47.2: Foundry Syslog Messages (Continued)

Message Level	Message	Explanation
Alert	MAC Authentication failed for <mac-address> on <portnum> (RADIUS given VLAN does not match with TAGGED vlan)	Multi-device port authentication failed for the <mac-address> on a tagged port because the packet with this MAC address as the source was tagged with a VLAN ID different from the RADIUS-supplied VLAN ID.
Alert	Management module at slot <slot-num> state changed from <module-state> to <module-state>.	Indicates a state change in a management module. The <slot-num> indicates the chassis slot containing the module. The <module-state> can be one of the following: • active • standby • crashed • coming-up • unknown
Alert	OSPF LSA Overflow, LSA Type = <lsa-type>	Indicates an LSA database overflow. The <lsa-type> parameter indicates the type of LSA that experienced the overflow condition. The LSA type is one of the following: • 1 – Router • 2 – Network • 3 – Summary • 4 – Summary • 5 – External
Alert	OSPF Memory Overflow	OSPF has run out of memory.
Alert	Power supply <num>, <location>, failed	A power supply has failed. The <num> is the power supply number. The <location> describes where the failed power supply is in the chassis.
Alert	System: Temperature is over shutdown level, system is going to be reset in <num> seconds	The chassis temperature has risen above shutdown level. The system will be shut down in the amount of time indicated.

Table 47.2: Foundry Syslog Messages (Continued)

Message Level	Message	Explanation
Alert	Temperature <degrees> C degrees, warning level <warn-degrees> C degrees, shutdown level <shutdown-degrees> C degrees	Indicates an over temperature condition on the active module. The <degrees> value indicates the temperature of the module. The <warn-degrees> value is the warning threshold temperature configured for the module. The <shutdown-degrees> value is the shutdown temperature configured for the module.
Critical	Authentication shut down <portnum> due to DOS attack	Denial of Service (DoS) attack protection was enabled for multi-device port authentication on the specified <portnum>, and the per-second rate of RADIUS authentication attempts for the port exceeded the configured limit. The Foundry device considers this to be a DoS attack and disables the port.
Debug	BGP4: Not enough memory available to run BGP4	The device could not start the BGP4 routing protocol because there is not enough memory available.
Debug	DOT1X: Not enough memory	There is not enough system memory for 802.1X authentication to take place. Contact Foundry Technical Support.
Error	No of prefixes received from BGP peer <ip-addr> exceeds maximum prefix-limit...shutdown	The Layer 3 Switch has received more than the specified maximum number of prefixes from the neighbor, and the Layer 3 Switch is therefore shutting down its BGP4 session with the neighbor.
Information	IPv6: IPv6 protocol disabled on the device from <session-id>	IPv6 protocol was disabled on the device during the specified session.
Information	IPv6: IPv6 protocol enabled on the device from <session-id>	IPv6 protocol was enabled on the device during the specified session.
Information	MAC Filter applied to port <port-id> by <username> from <session-id> (filter id=<filter-ids>)	Indicates a MAC filter was applied to the specified port by the specified user during the specified session. <session-id> can be console, telnet, ssh, web, or snmp. <filter-ids> is a list of the MAC filters that were applied.

Table 47.2: Foundry Syslog Messages (Continued)

Message Level	Message	Explanation
Information	MAC Filter removed from port <port-id> by <username> from <session-id> (filter id=<filter-ids>)	Indicates a MAC filter was removed from the specified port by the specified user during the specified session. <session-id> can be console, telnet, ssh, web, or snmp. <filter-ids> is a list of the MAC filters that were removed.
Information	Security: Password has been changed for user <username> from <session-id>	Password of the specified user has been changed during the specified session ID or type. <session-id> can be console, telnet, ssh, web, or snmp.
Informational	<device-name> : Logical link on interface ethernet <slot#/port#> is down.	The specified ports were logically brought down while **singleton** was configured on the port.
Informational	<device-name>: Logical link on interface ethernet <slot#/port#> is up.	The specified ports were logically brought up while **singleton** was configured on the port.
Informational	<user-name> login to PRIVILEGED mode	A user has logged into the Privileged EXEC mode of the CLI. The <user-name> is the user name.
Informational	<user-name> login to USER EXEC mode	A user has logged into the USER EXEC mode of the CLI. The <user-name> is the user name.
Informational	<user-name> logout from PRIVILEGED mode	A user has logged out of Privileged EXEC mode of the CLI. The <user-name> is the user name.
Informational	<user-name> logout from USER EXEC mode	A user has logged out of the USER EXEC mode of the CLI. The <user-name> is the user name.
Informational	ACL <acl id> added | deleted | modified from console | telnet | ssh | web | snmp session	A user created, modified, deleted, or applied an ACL via the Web, SNMP, console, SSH, or Telnet session.
Informational	Bridge is new root, vlan <vlan-id>, root ID <root-id>	A Spanning Tree Protocol (STP) topology change has occurred, resulting in the Foundry device becoming the root bridge. The <vlan-id> is the ID of the VLAN in which the STP topology change occurred. The <root-id> is the STP bridge root ID.

Table 47.2: Foundry Syslog Messages (Continued)

Message Level	Message	Explanation
Informational	Bridge root changed, vlan <vlan-id>, new root ID <string>, root interface <portnum>	A Spanning Tree Protocol (STP) topology change has occurred. The <vlan-id> is the ID of the VLAN in which the STP topology change occurred. The <root-id> is the STP bridge root ID. The <portnum> is the number of the port connected to the new root bridge.
Informational	Bridge topology change, vlan <vlan-id>, interface <portnum>, changed state to <stp-state>	A Spanning Tree Protocol (STP) topology change has occurred on a port. The <vlan-id> is the ID of the VLAN in which the STP topology change occurred. The <portnum> is the port number. The <stp-state> is the new STP state and can be one of the following: • disabled • blocking • listening • learning • forwarding • unknown
Informational	Cold start	The device has been powered on.
Informational	DOT1X : port <portnum> - mac <mac address> Cannot apply an ACL or MAC filter on a port member of a VE (virtual interface)	The RADIUS server returned an IP ACL or MAC address filter, but the port is a member of a virtual interface (VE).
Informational	DOT1X : port <portnum> - mac <mac address> cannot remove inbound ACL	An error occurred while removing the inbound ACL.
Informational	DOT1X : port <portnum> - mac <mac address> Downloading a MAC filter, but MAC filter have no effect on router port	The RADIUS server returned an MAC address filter, but the <portnum> is a router port (it has one or more IP addresses).
Informational	DOT1X : port <portnum> - mac <mac address> Downloading an IP ACL, but IP ACL have no effect on a switch port	The RADIUS server returned an IP ACL, but the <portnum> is a switch port (no IP address).
Informational	DOT1X : port <portnum> - mac <mac address> Error - could not add all MAC filters	The Foundry device was unable to implement the MAC address filters returned by the RADIUS server.
Informational	DOT1X : port <portnum> - mac <mac address> Invalid MAC filter ID - this ID doesn't exist	The MAC address filter ID returned by the RADIUS server does not exist in the Foundry device’s configuration.
Informational	DOT1X : port <portnum> - mac <mac address> Invalid MAC filter ID - this ID is user defined and cannot be used	The port was assigned a MAC address filter ID that had been dynamically created by another user.

Table 47.2: Foundry Syslog Messages (Continued)

Message Level	Message	Explanation
Informational	DOT1X : port <portnum> - mac <mac address> is unauthorized because system resource is not enough or the invalid information to set the dynamic assigned IP ACLs or MAC address filters	802.1X authentication failed for the Client with the specified <mac address> on the specified <portnum> either due to insufficient system resources on the device, or due to invalid IP ACL or MAC address filter information returned by the RADIUS server.
Informational	DOT1X : port <portnum> - mac <mac address> Port is already bound with MAC filter	The RADIUS server returned a MAC address filter, but a MAC address filter had already been applied to the port.
Informational	DOT1X : port <portnum> - mac <mac address> This device doesn't support ACL with MAC Filtering on the same port	The RADIUS server returned a MAC address filter while an IP ACL was applied to the port, or returned an IP ACL while a MAC address filter was applied to the port.
Informational	DOT1X Port <portnum> is unauthorized because system resource is not enough or the invalid information to set the dynamic assigned IP ACLs or MAC address filters	802.1X authentication could not take place on the port. This happened because strict security mode was enabled and one of the following occurred: • Insufficient system resources were available on the device to apply an IP ACL or MAC address filter to the port • Invalid information was received from the RADIUS server (for example, the Filter-ID attribute did not refer to an existing IP ACL or MAC address filter)
Informational	DOT1X: Port <portnum> currently used vlan-id changes to <vlan-id> due to dot1x-RADIUS vlan assignment	A user has completed 802.1X authentication. The profile received from the RADIUS server specifies a VLAN ID for the user. The port to which the user is connected has been moved to the VLAN indicated by <vlan-id>.
Informational	DOT1X: Port <portnum> currently used vlan-id is set back to port default vlan-id <vlan-id>	The user connected to <portnum> has disconnected, causing the port to be moved back into its default VLAN, <vlan-id>.
Informational	DOT1X: Port <portnum>, AuthControlledPortStatus change: authorized	The status of the interface's controlled port has changed from unauthorized to authorized.
Informational	DOT1X: Port <portnum>, AuthControlledPortStatus change: unauthorized	The status of the interface's controlled port has changed from authorized to unauthorized.
Informational	Enable super | port-config | read-only password deleted | added | modified from console | telnet | ssh | web | snmp OR Line password deleted | added | modified from console | telnet | ssh | web | snmp	A user created, re-configured, or deleted an Enable or Line password via the Web, SNMP, console, SSH, or Telnet session.
Informational	ERR_DISABLE: Interface ethernet <port-number>, err-disable recovery timeout	Errdisable recovery timer expired and the port has been reenabled.

Table 47.2: Foundry Syslog Messages (Continued)

Message Level	Message	Explanation
Informational	ERR_DISABLE: Interface ethernet 16, err-disable recovery timeout	If the wait time (port is down and is waiting to come up) expires and the port is brought up the following message is displayed.
Informational	ERR_DISABLE: Link flaps on port ethernet 16 exceeded threshold; port in err-disable state	The threshold for the number of times that a port's link toggles from "up" to "down" and "down" to "up" has been exceeded.
Informational	Interface <portnum>, line protocol down	The line protocol on a port has gone down. The <portnum> is the port number.
Informational	Interface <portnum>, line protocol up	The line protocol on a port has come up. The <portnum> is the port number.
Informational	Interface <portnum>, state down	A port has gone down. The <portnum> is the port number.
Informational	Interface <portnum>, state up	A port has come up. The <portnum> is the port number.
Informational	MAC Based Vlan Disabled on port <port id>	A MAC Based VLAN has been disabled on a port
Informational	MAC Based Vlan Enabled on port <port id>	A MAC Based VLAN has been enabled on a port.
Informational	MAC Filter added | deleted | modified from console | telnet | ssh | web | snmp session filter id = <MAC filter ID>, src mac = <Source MAC address> | any, dst mac = <Destination MAC address> | any	A user created, modified, deleted, or applied this MAC filter via the Web, SNMP, console, SSH, or Telnet session.
Informational	MSTP: BPDU-guard interface ethernet <port-number> detect (Received BPDU), putting into err-disable state.	BPDU guard violation occurred in MSTP.
Informational	Port <p> priority changed to <n>	A port's priority has changed.
Informational	Port <portnum>, srcip-security max-ipaddr-per-int reached.Last IP=<ipaddr>	The address limit specified by the **srcip-security max-ipaddr-per-interface** command has been reached for the port.
Informational	Port <portnum>, srcip-security max-ipaddr-per-int reached.Last IP=<ipaddr>	The address limit specified by the **srcip-security max-ipaddr-per-interface** command has been reached for the port.
Informational	Security: console login by <username> to USER | PRIVILEGE EXEC mode	The specified user logged into the device console into the specified EXEC mode.
Informational	Security: console logout by <username>	The specified user logged out of the device console.
Informational	Security: telnet | SSH login by <username> from src IP <ip-address>, src MAC <mac-address> to USER | PRIVILEGE EXEC mode	The specified user logged into the device using Telnet or SSH from the specified IP address and/or MAC address. The user logged into the specified EXEC mode.

Table 47.2: Foundry Syslog Messages (Continued)

Message Level	Message	Explanation
Informational	Security: telnet | SSH logout by <username> from src IP <ip-address>, src MAC <mac-address> to USER | PRIVILEGE EXEC mode	The specified user logged out of the device. The user was using Telnet or SSH to access the device from the specified IP address and/ or MAC address. The user logged out of the specified EXEC mode.
Informational	SNMP read-only community | read-write community | contact | location | user | group | view | engineld | trap [host] [<value -str>] deleted | added | modified from console | telnet | ssh | web | snmp session	A user made SNMP configuration changes via the Web, SNMP, console, SSH, or Telnet session. [<value-str>] does not appear in the message if SNMP **community** or **engineId** is specified.
Informational	SNMP Auth. failure, intruder IP: <ip-addr>	A user has tried to open a management session with the device using an invalid SNMP community string. The <ip-addr> is the IP address of the host that sent the invalid community string.
Informational	SSH | telnet server enabled | disabled from console | telnet | ssh | web | snmp session [by user <username>]	A user enabled or disabled an SSH or Telnet session, or changed the SSH enable/disable configuration via the Web, SNMP, console, SSH, or Telnet session.
Informational	startup-config was changed or startup-config was changed by <user-name>	A configuration change was saved to the startup-config file. The <user-name> is the user's ID, if they entered a user ID to log in.
Informational	STP: Root Guard Port <port-number>, VLAN <vlan-ID> consistent (Timeout).	Root guard unblocks a port.
Informational	STP: Root Guard Port <port-number>, VLAN <vlan-ID> inconsistent (Received superior BPDU).	Root guard blocked a port.
Informational	STP: VLAN <vlan id> BPDU-Guard on Port <port id> triggered (Received BPDU), putting into err-disable state	The BPDU guard feature has detected an incoming BPDU on {vlan-id, port-id}
Informational	STP: VLAN <vlan id> Root-Protect Port <port id>, Consistent (Timeout)	The root protect feature goes back to the consistent state.
Informational	STP: VLAN <vlan id> Root-Protect Port <port id>, Inconsistent (Received superior BPDU)	The root protect feature has detected a superior BPDU and goes into the inconsistent state on {vlan-id, port-id}.
Informational	STP: VLAN <vlan-id> BPDU-guard port <port-number> detect (Received BPDU), putting into err-disable state	STP placed a port into an errdisable state for BPDU guard.
Informational	STP: VLAN 1 BPDU-guard port <port-number> detect (Received BPDU), putting into err-disable state.	BPDU guard violation in occurred in STP or RSTP.

Table 47.2: Foundry Syslog Messages (Continued)

Message Level	Message	Explanation
Informational	Syslog server <IP-address> deleted | added | modified from console | telnet | ssh | web | snmp OR Syslog operation enabled | disabled from console | telnet | ssh | web | snmp	A user made Syslog configuration changes to the specified Syslog server address, or enabled or disabled a Syslog operation via the Web, SNMP, console, SSH, or Telnet session.
Informational	System: Fan <fan id> (from left when facing right side), ok	The fan status has changed from fail to normal.
Informational	System: Fan speed changed automatically to <fan speed>	The system automatically changed the fan speed to the speed specified in this message.
Informational	telnet | SSH | web access [by <username>] from src IP <source ip address>, src MAC <source MAC address> rejected, <n> attempt(s)	There were failed web, SSH, or Telnet login access attempts from the specified source IP and MAC address. • [by <user> <username>] does not appear if **telnet** or **SSH** clients are specified. • <n> is the number of times this SNMP trap occurred in the last five minutes, or other configured number of minutes.
Informational	Trunk group (<ports>) created by 802.3ad link-aggregation module.	802.3ad link aggregation is configured on the device, and the feature has dynamically created a trunk group (aggregate link). The <ports> is a list of the ports that were aggregated to make the trunk group.
Informational	user <username> added | deleted | modified from console | telnet | ssh | web | snmp	A user created, modified, or deleted a local user account via the Web, SNMP, console, SSH, or Telnet session.
Informational	vlan <vlan id> added | deleted | modified from console | telnet | ssh | web | snmp session	A user created, modified, or deleted a VLAN via the Web, SNMP, console, SSH, or Telnet session.
Informational	Warm start	The system software (flash code) has been reloaded.
Informational	vlan <vlan-id> Bridge is RootBridge <mac-address> (MgmtPriChg)	802.1W changed the current bridge to be the root bridge of the given topology due to administrative change in bridge priority.
Informational	vlan <vlan-id> Bridge is RootBridge <mac-address> (MsgAgeExpiry)	The message age expired on the Root port so 802.1W changed the current bridge to be the root bridge of the topology.
Informational	vlan <vlan-id> interface <portnum> Bridge TC Event (DOT1wTransition)	802.1W recognized a topology change event in the bridge. The topology change event is the forwarding action that started on a non-edge Designated port or Root port.

Table 47.2: Foundry Syslog Messages (Continued)

Message Level	Message	Explanation
Informational	vlan <vlan-id> interface <portnum> STP state -> <state> (DOT1wTransition)	802.1W changed the state of a port to a new state: forwarding, learning, blocking. If the port changes to blocking, the bridge port is in discarding state.
Informational	vlan <vlan-id> New RootBridge <mac-address> RootPort <portnum> (BpduRcvd)	802.1W selected a new root bridge as a result of the BPDUs received on a bridge port.
Informational	vlan <vlan-id> New RootPort <portnum> (RootSelection)	802.1W changed the port's role to Root port, using the root selection computation.
Informational	OPTICAL MONITORING: port <port-number> is not capable.	The optical transceiver is qualified by Foundry, but the transceiver does not support digital optical performance monitoring.
Informational	SYSTEM: Optic is not Foundry-qualified (<port-number>)	Foundry does not support the optical transceiver. **NOTE:** Foundry does not support optical transceivers manufactured by other vendors.
Notification	ACL exceed max DMA L4 cam resource, using flow based ACL instead	The port does not have enough Layer 4 CAM entries for the ACL. To correct this condition, allocate more Layer 4 CAM entries. To allocate more Layer 4 CAM entries, enter the following command at the CLI configuration level for the interface: **ip access-group max-l4-cam** <num>
Notification	ACL insufficient L4 cam resource, using flow based ACL instead	The port does not have a large enough CAM partition for the ACLs.
Notification	ACL insufficient L4 session resource, using flow based ACL instead	The device does not have enough Layer 4 session entries. To correct this condition, allocate more memory for sessions. To allocate more memory, enter the following command at the global CONFIG level of the CLI interface: **system-max session-limit** <num>
Notification	ACL port fragment packet inspect rate <rate> exceeded on port <portnum>	The fragment rate allowed on an individual interface has been exceeded. The <rate> indicates the maximum rate allowed. The <portnum> indicates the port. This message can occur if fragment thottling is enabled.

Table 47.2: Foundry Syslog Messages (Continued)

Message Level	Message	Explanation
Notification	ACL system fragment packet inspect rate <rate> exceeded	The fragment rate allowed on the device has been exceeded. The <rate> indicates the maximum rate allowed. This message can occur if fragment thottling is enabled.
Notification	Authentication Disabled on <portnum>	The multi-device port authentication feature was disabled on the on the specified <portnum>.
Notification	Authentication Enabled on <portnum>	The multi-device port authentication feature was enabled on the on the specified <portnum>.
Notification	BGP Peer <ip-addr> DOWN (IDLE)	Indicates that a BGP4 neighbor has gone down. The <ip-addr> is the IP address of the neighbor's BGP4 interface with the Foundry device.
Notification	BGP Peer <ip-addr> UP (ESTABLISHED)	Indicates that a BGP4 neighbor has come up. The <ip-addr> is the IP address of the neighbor's BGP4 interface with the Foundry device.
Notification	DOT1X issues software but not physical port down indication of Port <portnum> to other software applications	The device has indicated that the specified is no longer authorized, but the actual port may still be active.
Notification	DOT1X issues software but not physical port up indication of Port <portnum> to other software applications	The device has indicated that the specified port has been authenticated, but the actual port may not be active.
Notification	DOT1X: Port <port_id> Mac <mac_address> -user <user_id> - RADIUS timeout for authentication	The RADIUS session has timed out for this 802.1x port.
Notification	ISIS ENTERED INTO OVERLOAD STATE	The Layer 3 Switch has set the overload bit to on (1), indicating that the Layer 3 Switch's IS-IS resources are overloaded.
Notification	ISIS EXITING FROM OVERLOAD STATE	The Layer 3 Switch has set the overload bit to off (0), indicating that the Layer 3 Switch's IS-IS resources are no longer overloaded.
Notification	ISIS L1 ADJACENCY DOWN <system-id> on circuit <circuit-id>	The Layer 3 Switch's adjacency with this Level-1 IS has gone down. The <system-id> is the system ID of the IS. The <circuit-id> is the ID of the circuit over which the adjacency was established.

Table 47.2: Foundry Syslog Messages (Continued)

Message Level	Message	Explanation
Notification	ISIS L1 ADJACENCY UP <system-id> on circuit <circuit-id>	The Layer 3 Switch's adjacency with this Level-1 IS has come up. The <system-id> is the system ID of the IS. The <circuit-id> is the ID of the circuit over which the adjacency was established.
Notification	ISIS L2 ADJACENCY DOWN <system-id> on circuit <circuit-id>	The Layer 3 Switch's adjacency with this Level-2 IS has gone down. The <system-id> is the system ID of the IS. The <circuit-id> is the ID of the circuit over which the adjacency was established.
Notification	ISIS L2 ADJACENCY UP <system-id> on circuit <circuit-id>	The Layer 3 Switch's adjacency with this Level-2 IS has come up. The <system-id> is the system ID of the IS. The <circuit-id> is the ID of the circuit over which the adjacency was established.
Notification	Local ICMP exceeds <burst-max> burst packets, stopping for <lockup> seconds!!	The number of ICMP packets exceeds the <burst-max> threshold set by the **ip icmp burst** command. The Foundry device may be the victim of a Denial of Service (DoS) attack. All ICMP packets will be dropped for the number of seconds specified by the <lockup> value. When the lockup period expires, the packet counter is reset and measurement is restarted.
Notification	Local TCP exceeds <burst-max> burst packets, stopping for <lockup> seconds!!	The number of TCP SYN packets exceeds the <burst-max> threshold set by the **ip tcp burst** command. The Foundry device may be the victim of a TCP SYN DoS attack. All TCP SYN packets will be dropped for the number of seconds specified by the <lockup> value. When the lockup period expires, the packet counter is reset and measurement is restarted.
Notification	Local TCP exceeds <num> burst packets, stopping for <num> seconds!!	Threshold parameters for local TCP traffic on the device have been configured, and the maximum burst size for TCP packets has been exceeded. The first <num> is the maximum burst size (maximum number of packets allowed). The second <num> is the number of seconds during which additional TCP packets will be blocked on the device. **Note**: This message can occur in response to an attempted TCP SYN attack.

Table 47.2: Foundry Syslog Messages (Continued)

Message Level	Message	Explanation
Notification	MAC Authentication succeeded for <mac-address> on <portnum>	RADIUS authentication was successful for the specified <mac-address> on the specified <portnum>.
Notification	MAC Authentication RADIUS timeout for <mac_address> on port <port_id>	The RADIUS session has timed out for the MAC address for this port.
Notification	Module was inserted to slot <slot-num>	Indicates that a module was inserted into a chassis slot. The <slot-num> is the number of the chassis slot into which the module was inserted.
Notification	Module was removed from slot <slot-num>	Indicates that a module was removed from a chassis slot. The <slot-num> is the number of the chassis slot from which the module was removed.
Notification	OSPF interface state changed, rid <router-id>, intf addr <ip-addr>, state <ospf-state>	Indicates that the state of an OSPF interface has changed. The <router-id> is the router ID of the Foundry device. The <ip-addr> is the interface's IP address. The <ospf-state> indicates the state to which the interface has changed and can be one of the following: • down • loopback • waiting • point-to-point • designated router • backup designated router • other designated router • unknown

Table 47.2: Foundry Syslog Messages (Continued)

Message Level	Message	Explanation
Notification	OSPF intf authen failure, rid <router-id>, intf addr <ip-addr>, pkt src addr <src-ip-addr>, error type <error-type>, pkt type <pkt-type>	Indicates that an OSPF interface authentication failure has occurred. The <router-id> is the router ID of the Foundry device. The <ip-addr> is the IP address of the interface on the Foundry device. The <src-ip-addr> is the IP address of the interface from which the Foundry device received the authentication failure. The <error-type> can be one of the following: • bad version • area mismatch • unknown NBMA neighbor • unknown virtual neighbor • authentication type mismatch • authentication failure • network mask mismatch • hello interval mismatch • dead interval mismatch • option mismatch • unknown The <packet-type> can be one of the following: • hello • database description • link state request • link state update • link state ack • unknown

Table 47.2: Foundry Syslog Messages (Continued)

Message Level	Message	Explanation
Notification	OSPF intf config error, rid <router-id>, intf addr <ip-addr>, pkt src addr <src-ip-addr>, error type <error-type>, pkt type <pkt-type>	Indicates that an OSPF interface configuration error has occurred. The <router-id> is the router ID of the Foundry device. The <ip-addr> is the IP address of the interface on the Foundry device. The <src-ip-addr> is the IP address of the interface from which the Foundry device received the error packet. The <error-type> can be one of the following: • bad version • area mismatch • unknown NBMA neighbor • unknown virtual neighbor • authentication type mismatch • authentication failure • network mask mismatch • hello interval mismatch • dead interval mismatch • option mismatch • unknown The <packet-type> can be one of the following: • hello • database description • link state request • link state update • link state ack • unknown

Table 47.2: Foundry Syslog Messages (Continued)

Message Level	Message	Explanation
Notification	OSPF intf rcvd bad pkt, rid <router-id>, intf addr <ip-addr>, pkt src addr <src-ip-addr>, pkt type <pkt-type>	Indicates that an OSPF interface received a bad packet. The <router-id> is the router ID of the Foundry device. The <ip-addr> is the IP address of the interface on the Foundry device. The <src-ip-addr> is the IP address of the interface from which the Foundry device received the authentication failure. The <packet-type> can be one of the following: • hello • database description • link state request • link state update • link state ack • unknown
Notification	OSPF intf rcvd bad pkt: Bad Checksum, rid <ip-addr>, intf addr <ip-addr>, pkt size <num>, checksum <num>, pkt src addr <ip-addr>, pkt type <type>	The device received an OSPF packet that had an invalid checksum. The rid <ip-addr> is Foundry device’s router ID. The intf addr <ip-addr> is the IP address of the Foundry interface that received the packet. The pkt size <num> is the number of bytes in the packet. The checksum <num> is the checksum value for the packet. The pkt src addr <ip-addr> is the IP address of the neighbor that sent the packet. The pkt type <type> is the OSPF packet type and can be one of the following: • hello • database description • link state request • link state update • link state acknowledgement • unknown (indicates an invalid packet type)

Table 47.2: Foundry Syslog Messages (Continued)

Message Level	Message	Explanation
Notification	OSPF intf rcvd bad pkt: Bad Packet type, rid <ip-addr>, intf addr <ip-addr>, pkt size <num>, checksum <num>, pkt src addr <ip-addr>, pkt type <type>	The device received an OSPF packet with an invalid type. The parameters are the same as for the Bad Checksum message. The pkt type <type> value is “unknown”, indicating that the packet type is invalid.
Notification	OSPF intf rcvd bad pkt: Invalid packet size, rid <ip-addr>, intf addr <ip-addr>, pkt size <num>, checksum <num>, pkt src addr <ip-addr>, pkt type <type>	The device received an OSPF packet with an invalid packet size. The parameters are the same as for the Bad Checksum message.
Notification	OSPF intf rcvd bad pkt: Unable to find associated neighbor, rid <ip-addr>, intf addr <ip-addr>, pkt size <num>, checksum <num>, pkt src addr <ip-addr>, pkt type <type>	The neighbor IP address in the packet is not on the Foundry device’s list of OSPF neighbors. The parameters are the same as for the Bad Checksum message.
Notification	OSPF intf retransmit, rid <router-id>, intf addr <ip-addr>, nbr rid <nbr-router-id>, pkt type is <pkt-type>, LSA type <lsa-type>, LSA id <lsa-id>, LSA rid <lsa-router-id>	An OSPF interface on the Foundry device has retransmitted a Link State Advertisement (LSA). The <router-id> is the router ID of the Foundry device. The <ip-addr> is the IP address of the interface on the Foundry device. The <nbr-router-id> is the router ID of the neighbor router. The <packet-type> can be one of the following: • hello • database description • link state request • link state update • link state ack • unknown The <lsa-type> is the type of LSA. The <lsa-id> is the LSA ID. The <lsa-router-id> is the LSA router ID.
Notification	OSPF LSDB approaching overflow, rid <router-id>, limit <num>	The software is close to an LSDB condition. The <router-id> is the router ID of the Foundry device. The <num> is the number of LSAs.

Table 47.2: Foundry Syslog Messages (Continued)

Message Level	Message	Explanation
Notification	OSPF LSDB overflow, rid <router-id>, limit <num>	A Link State Database Overflow (LSDB) condition has occurred. The <router-id> is the router ID of the Foundry device. The <num> is the number of LSAs.
Notification	OSPF max age LSA, rid <router-id>, area <area-id>, LSA type <lsa-type>, LSA id <lsa-id>, LSA rid <lsa-router-id>	An LSA has reached its maximum age. The <router-id> is the router ID of the Foundry device. The <area-id> is the OSPF area. The <lsa-type> is the type of LSA. The <lsa-id> is the LSA ID. The <lsa-router-id> is the LSA router ID.
Notification	OSPF nbr state changed, rid <router-id>, nbr addr <ip-addr>, nbr rid <nbr-router-Id>, state <ospf-state>	Indicates that the state of an OSPF neighbor has changed. The <router-id> is the router ID of the Foundry device. The <ip-addr> is the IP address of the neighbor. The <nbr-router-id> is the router ID of the neighbor. The <ospf-state> indicates the state to which the interface has changed and can be one of the following: • down • attempt • initializing • 2-way • exchange start • exchange • loading • full • unknown

Table 47.2: Foundry Syslog Messages (Continued)

Message Level	Message	Explanation
Notification	OSPF originate LSA, rid <router-id>, area <area-id>, LSA type <lsa-type>, LSA id <lsa-id>, LSA router id <lsa-router-id>	An OSPF interface has originated an LSA. The <router-id> is the router ID of the Foundry device. The <area-id> is the OSPF area. The <lsa-type> is the type of LSA. The <lsa-id> is the LSA ID. The <lsa-router-id> is the LSA router ID.
Notification	OSPF virtual intf authen failure, rid <router-id>, intf addr <ip-addr>, pkt src addr <src-ip-addr>, error type <error-type>, pkt type <pkt-type>	Indicates that an OSPF virtual routing interface authentication failure has occurred. The <router-id> is the router ID of the Foundry device. The <ip-addr> is the IP address of the interface on the Foundry device. The <src-ip-addr> is the IP address of the interface from which the Foundry device received the authentication failure. The <error-type> can be one of the following: • bad version • area mismatch • unknown NBMA neighbor • unknown virtual neighbor • authentication type mismatch • authentication failure • network mask mismatch • hello interval mismatch • dead interval mismatch • option mismatch • unknown The <packet-type> can be one of the following: • hello • database description • link state request • link state update • link state ack • unknown

Table 47.2: Foundry Syslog Messages (Continued)

Message Level	Message	Explanation
Notification	OSPF virtual intf config error, rid <router-id>, intf addr <ip-addr>, pkt src addr <src-ip-addr>, error type <error-type>, pkt type <pkt-type>	Indicates that an OSPF virtual routing interface configuration error has occurred. The <router-id> is the router ID of the Foundry device. The <ip-addr> is the IP address of the interface on the Foundry device. The <src-ip-addr> is the IP address of the interface from which the Foundry device received the error packet. The <error-type> can be one of the following: • bad version • area mismatch • unknown NBMA neighbor • unknown virtual neighbor • authentication type mismatch • authentication failure • network mask mismatch • hello interval mismatch • dead interval mismatch • option mismatch • unknown The <packet-type> can be one of the following: • hello • database description • link state request • link state update • link state ack • unknown

Table 47.2: Foundry Syslog Messages (Continued)

Message Level	Message	Explanation
Notification	OSPF virtual intf rcvd bad pkt, rid <router-id>, intf addr <ip-addr>, pkt src addr <src-ip-addr>, pkt type <pkt-type>	Indicates that an OSPF interface received a bad packet. The <router-id> is the router ID of the Foundry device. The <ip-addr> is the IP address of the interface on the Foundry device. The <src-ip-addr> is the IP address of the interface from which the Foundry device received the authentication failure. The <packet-type> can be one of the following: • hello • database description • link state request • link state update • link state ack • unknown
Notification	OSPF virtual intf retransmit, rid <router-id>, intf addr <ip-addr>, nbr rid <nbr-router-id>, pkt type is <pkt-type>, LSA type <lsa-type>, LSA id <lsa-id>, LSA rid <lsa-router-id>	An OSPF interface on the Foundry device has retransmitted a Link State Advertisement (LSA). The <router-id> is the router ID of the Foundry device. The <ip-addr> is the IP address of the interface on the Foundry device. The <nbr-router-id> is the router ID of the neighbor router. The <packet-type> can be one of the following: • hello • database description • link state request • link state update • link state ack • unknown The <lsa-type> is the type of LSA. The <lsa-id> is the LSA ID. The <lsa-router-id> is the LSA router ID.

Table 47.2: Foundry Syslog Messages (Continued)

Message Level	Message	Explanation
Notification	OSPF virtual intf state changed, rid <router-id>, area <area-id>, nbr <ip-addr>, state <ospf-state>	Indicates that the state of an OSPF virtual routing interface has changed. The <router-id> is the router ID of the router the interface is on. The <area-id> is the area the interface is in. The <ip-addr> is the IP address of the OSPF neighbor. The <ospf-state> indicates the state to which the interface has changed and can be one of the following: • down • loopback • waiting • point-to-point • designated router • backup designated router • other designated router • unknown
Notification	OSPF virtual nbr state changed, rid <router-id>, nbr addr <ip-addr>, nbr rid <nbr-router-id>, state <ospf-state>	Indicates that the state of an OSPF virtual neighbor has changed. The <router-id> is the router ID of the Foundry device. The <ip-addr> is the IP address of the neighbor. The <nbr-router-id> is the router ID of the neighbor. The <ospf-state> indicates the state to which the interface has changed and can be one of the following: • down • attempt • initializing • 2-way • exchange start • exchange • loading • full • unknown

Table 47.2: Foundry Syslog Messages (Continued)

Message Level	Message	Explanation
Notification	Transit ICMP in interface <portnum> exceeds <num> burst packets, stopping for <num> seconds!!	Threshold parameters for ICMP transit (through) traffic have been configured on an interface, and the maximum burst size for ICMP packets on the interface has been exceeded. The <portnum> is the port number. The first <num> is the maximum burst size (maximum number of packets allowed). The second <num> is the number of seconds during which additional ICMP packets will be blocked on the interface. **Note**: This message can occur in response to an attempted Smurf attack.
Notification	Transit TCP in interface <portnum> exceeds <num> burst packets, stopping for <num> seconds!	Threshold parameters for TCP transit (through) traffic have been configured on an interface, and the maximum burst size for TCP packets on the interface has been exceeded. The <portnum> is the port number. The first <num> is the maximum burst size (maximum number of packets allowed). The second <num> is the number of seconds during which additional TCP packets will be blocked on the interface. **Note**: This message can occur in response to an attempted TCP SYN attack.
Notification	VRRP intf state changed, intf <portnum>, vrid <virtual-router-id>, state <vrrp-state>	A state change has occurred in a Virtual Router Redundancy Protocol (VRRP) interface. The <portnum> is the port. The <virtual-router-id> is the virtual router ID (VRID) configured on the interface. The <vrrp-state> can be one of the following: • init • master • backup • unknown
Warning	DOT1X security violation at port <portnum>, malicious mac address detected: <mac-address>	A security violation was encountered at the specified port number.

Table 47.2: Foundry Syslog Messages (Continued)

Message Level	Message	Explanation
Warning	Dup IP <ip-addr> detected, sent from MAC <mac-addr> interface <portnum>	Indicates that the Foundry device received a packet from another device on the network with an IP address that is also configured on the Foundry device. The <ip-addr> is the duplicate IP address. The <mac-addr> is the MAC address of the device with the duplicate IP address. The <portnum> is the Foundry port that received the packet with the duplicate IP address. The address is the packet's source IP address.
Warning	IGMP/MLD no hardware vidx, broadcast to the entire vlan. rated limited number	IGMP or MLD snooping has run out of hardware application VLANs. There are 4096 application VLANs per device. Traffic streams for snooping entries without an application VLAN are switched to the entire VLAN and to the CPU to be dropped. This message is rate-limited to appear a maximum of once every 10 minutes. The rate-limited number shows the number on non-printed warnings.
Warning	IGMP/MLD: <vlanId>(<portId>) is V1 but rcvd V2 from nbr <ipAddr>	Port has received a query with a MLD version that does not match the port's MLD version. This message is rated-limited to appear a maximum of once every 10 hours.
Warning	list <acl-num> denied <ip-proto> <src-ip-addr> (<src-tcp/udp-port>) (Ethernet <portnum> <mac-addr>) -> <dst-ip-addr> (<dst-tcp/udp-port>), 1 event(s)	Indicates that an Access Control List (ACL) denied (dropped) packets. The <acl-num> indicates the ACL number. Numbers 1 – 99 indicate standard ACLs. Numbers 100 – 199 indicate extended ACLs. The <ip-proto> indicates the IP protocol of the denied packets. The <src-ip-addr> is the source IP address of the denied packets. The <src-tcp/udp-port> is the source TCP or UDP port, if applicable, of the denied packets. The <portnum> indicates the port number on which the packet was denied. The <mac-addr> indicates the source MAC address of the denied packets. The <dst-ip-addr> indicates the destination IP address of the denied packets. The <dst-tcp/udp-port> indicates the destination TCP or UDP port number, if applicable, of the denied packets.

Table 47.2: Foundry Syslog Messages (Continued)

Message Level	Message	Explanation
Warning	Locked address violation at interface e<portnum>, address <mac-address>	Indicates that a port on which you have configured a lock-address filter received a packet that was dropped because the packet's source MAC address did not match an address learned by the port before the lock took effect. The e<portnum> is the port number. The <mac-address> is the MAC address that was denied by the address lock. Assuming that you configured the port to learn only the addresses that have valid access to the port, this message indicates a security violation.
Warning	mac filter group denied packets on port <portnum> src macaddr <mac-addr>, <num> packets	Indicates that a Layer 2 MAC filter group configured on a port has denied packets. The <portnum> is the port on which the packets were denied. The <mac-addr> is the source MAC address of the denied packets. The <num> indicates how many packets matching the values above were dropped during the five-minute interval represented by the log entry.
Warning	multicast no software resource: resource-name, rate limited number	IGMP or MLD snooping has run out of software resources. This message is rate-limited to appear a maximum of once every 10 minutes. The rate-limited number shows the number of non-printed warnings.
Warning	No global IP! cannot send IGMP msg.	The device is configured for **ip multicast active** but there is no configured IP address and the device cannot send out IGMP queries.
Warning	No of prefixes received from BGP peer <ip-addr> exceeds warning limit <num>	The Layer 3 Switch has received more than the allowed percentage of prefixes from the neighbor. The <ip-addr> is the IP address of the neighbor. The <num> is the number of prefixes that matches the percentage you specified. For example, if you specified a threshold of 100 prefixes and 75 percent as the warning threshold, this message is generated if the Layer 3 Switch receives a 76th prefix from the neighbor.

Table 47.2: Foundry Syslog Messages (Continued)

Message Level	Message	Explanation
Warning	NTP server <ip-addr> failed to respond	Indicates that a Simple Network Time Protocol (SNTP) server did not respond to the device's query for the current time. The <ip-addr> indicates the IP address of the SNTP server.
Warning	rip filter list <list-num> <direction> V1 | V2 denied <ip-addr>, <num> packets	Indicates that a RIP route filter denied (dropped) packets. The <list-num> is the ID of the filter list. The <direction> indicates whether the filter was applied to incoming packets or outgoing packets. The value can be one of the following: • in • out The V1 or V2 value specifies the RIP version (RIPv1 or RIPv2). The <ip-addr> indicates the network number in the denied updates. The <num> indicates how many packets matching the values above were dropped during the five-minute interval represented by the log entry.
Warning	Temperature is over warning level.	The chassis temperature has risen above the warning level.
Warning	Latched low RX Power | TX Power | TX Bias Current | Supply Voltage | Temperature warning alarm | warning, port <port-number>	The optical transceiver on the given port has risen above or fallen below the alarm or warning threshold.

Appendix B
Network Monitoring

This appendix describes the remote monitoring features available on Foundry products:

Basic Management

The following sections contain procedures for basic system management tasks.

Viewing System Information

You can access software and hardware specifics for a Foundry Layer 2 Switch or Layer 3 Switch.

To view the software and hardware details for the system, enter the **show version** command:

```
FastIron#show version
```

Syntax: show version

Viewing Configuration Information

You can view a variety of configuration details and statistics with the show option. The **show** option provides a convenient way to check configuration changes before saving them to flash.

The show options available will vary for Layer 2 Switches and Layer 3 Switches and by configuration level.

To determine the available show commands for the system or a specific level of the CLI, enter the following command:

```
FastIron#show ?
```

Syntax: show <option>

You also can enter “show” at the command prompt, then press the TAB key.

Viewing Port Statistics

Port statistics are polled by default every 10 seconds.

You can view statistics for ports by entering the following **show** commands:

- show interfaces
- show configuration
- show statistics

To display the statistics, enter a command such as the following:

```
FastIron#show statistics ethernet 1/3
Port  Link State     Dupl Speed Trunk Tag Priori MAC            Name
1/3    Up   Forward   Half 100M  None  No  level0 00e0.5200.0102

 Port 1/3 Counters:
        InOctets                3200          OutOctets                256
          InPkts                  50            OutPkts                  4
  InBroadcastPkts                  0    OutBroadcastPkts                  3
  InMulticastPkts                 48    OutMulticastPkts                  0
    InUnicastPkts                  2      OutUnicastPkts                  1
        InBadPkts                  0
      InFragments                  0
       InDiscards                  0          OutErrors                  0
              CRC                  0         Collisions                  0
         InErrors                  0     LateCollisions                  0
      InGiantPkts                  0
      InShortPkts                  0
         InJabber                  0
   InFlowCtrlPkts                  0     OutFlowCtrlPkts                  0
     InBitsPerSec                264       OutBitsPerSec                 16
     InPktsPerSec                  0       OutPktsPerSec                  0
     InUtilization              0.00%      OutUtilization               0.00%
```

Syntax: show statistics [ethernet [<stacknum>/<slotnum>/]<portnum>]

The <stacknum> parameter is required on FastIron GS devices running software release 02.5.00 or later.

The <slotnum> parameter is required on chassis devices and on FastIron GS devices running software release 02.5.00 or later.

The <portnum> parameter is a valid port number.

Table B.1 lists the statistics displayed in the output of the **show statistics** command.

Table B.1: Port Statistics

This Line...	Displays...
Port Configuration	
Port	The port number.
Link	The link state.
State	The STP state.
Dupl	The mode (full-duplex or half-duplex).
Speed	The port speed (10M, 100M, or 1000M).
Trunk	The trunk group number, if the port is a member of a trunk group.
Tag	Whether the port is a tagged member of a VLAN.
Priori	The QoS forwarding priority of the port (level0 – level7).
MAC	The MAC address of the port.

Table B.1: Port Statistics (Continued)

This Line...	Displays...
Name	The name of the port, if you assigned a name.
Statistics	
InOctets	The total number of good octets and bad octets received.
OutOctets	The total number of good octets and bad octets sent.
InPkts	The total number of packets received. The count includes rejected and local packets that are not sent to the switching core for transmission.
OutPkts	The total number of good packets sent. The count includes unicast, multicast, and broadcast packets.
InBroadcastPkts	The total number of good broadcast packets received.
OutBroadcastPkts	The total number of good broadcast packets sent.
InMulticastPkts	The total number of good multicast packets received.
OutMulticastPkts	The total number of good multicast packets sent.
InUnicastPkts	The total number of good unicast packets received.
OutUnicastPkts	The total number of good unicast packets sent.
InBadPkts	The total number of packets received for which one of the following is true: • The CRC was invalid. • The packet was oversized. • Jabbers: The packets were longer than 1518 octets and had a bad FCS. • Fragments: The packets were less than 64 octets long and had a bad FCS. • The packet was undersized (short).
InFragments	The total number of packets received for which both of the following was true: • The length was less than 64 bytes. • The CRC was invalid.
InDiscards	The total number of packets that were received and then dropped due to a lack of receive buffers.
OutErrors	The total number of packets with internal transmit errors such as TX underruns.
CRC	The total number of packets received for which all of the following was true: • The data length was between 64 bytes and the maximum allowable frame size. • No Collision or Late Collision was detected. • The CRC was invalid.
Collisions	The total number of packets received in which a Collision event was detected.
InErrors	The total number of packets received that had Alignment errors or phy errors.
LateCollisions	The total number of packets received in which a Collision event was detected, but for which a receive error (Rx Error) event was not detected.

Table B.1: Port Statistics (Continued)

This Line...	Displays...
InGiantPkts	The total number of packets for which all of the following was true: • The data length was longer than the maximum allowable frame size. • No Rx Error was detected. **Note**: Packets are counted for this statistic regardless of whether the CRC is valid or invalid.
InShortPkts	The total number of packets received for which all of the following was true: • The data length was less than 64 bytes. • No Rx Error was detected. • No Collision or Late Collision was detected. **Note**: Packets are counted for this statistic regardless of whether the CRC is valid or invalid.
InJabber	The total number of packets received for which all of the following was true: • The data length was longer than the maximum allowable frame size. • No Rx Error was detected. • The CRC was invalid.
InFlowCtrlPkts	The total number of flow control packets received.
OutFlowCtrlPkts	The total number of flow control packets transmitted.
InBitsPerSec	The number of bits received per second.
OutBitsPerSec	The number of bits sent per second.
InPktsPerSec	The number of packets received per second.
OutPktsPerSec	The number of packets sent per second.
InUtilization	The percentage of the port’s bandwidth used by received traffic.
OutUtilization	The percentage of the port’s bandwidth used by sent traffic.

Viewing STP Statistics

You can view a summary of STP statistics for Layer 2 Switches and Layer 3 Switches. STP statistics are by default polled every 10 seconds.

To view spanning tree statistics, enter the **show span** command. To view STP statistics for a VLAN, enter the **span vlan** command.

Clearing Statistics

You can clear statistics for many parameters with the clear option.

To determine the available **clear** commands for the system, enter the following command:

```
FastIron#clear ?
```

Syntax: clear <option>

You also can enter “clear” at the command prompt, then press the TAB key.

NOTE: Clear commands are found at the Privileged EXEC level.

Traffic Counters for Outbound Traffic

Platform Support:

- FESX/FSX/FWSX devices running software release 02.5.00 and later

You can configure traffic counters (also called transmit counters) that enable the Foundry device to count the following packet types on a port or port region:

- broadcast packets
- multicast packets
- unicast packets
- dropped packets due to congestion and egress filtering

Depending on the parameters specified with the traffic counter configuration, traffic counters record the number of outbound packets from any combination of the following sources:

- a specific port or all ports in a specific port region
- a specific VLAN or all VLANs
- a specific 802.1p priority queue or all priority queues

Configuration Notes

- This feature is supported in the Layer 2, base Layer 3, and full Layer 3 codes.
- This feature is not supported on the FGS.
- This feature applies to physical ports only, including 10-Gigabit Ethernet ports and trunk ports. It does not apply to virtual interfaces.
- Once the enhanced traffic counters are read using the **show transmit-counter values** command, the counters are cleared (reset to zero).
- For each port region, you can enable a maximum of two traffic counters, regardless of whether traffic counters are enabled on individual ports or on all ports in the port region.
- Traffic counters increase for bridged filtered outbound traffic when any of the following conditions occur:
    - The port is disabled or the link is down.
    - The port or port region does not belong to the VLAN specified in the transmit counter configuration.
    - A Layer 2 protocol (e.g., spanning tree) has the port in a Blocked state.
    - The source port needs to be suppressed for multi-target packets.
    - The priority queue specified in the traffic counter is not allowed for some other reason.
    - Unknown unicast and unregistered multicast packets are filtered.

Configuration Syntax

This section provides the syntax and configuration examples for enhanced traffic counters.

EXAMPLES:

To configure traffic counters for outbound traffic on a specific port, enter a command such as the following:

```
FastIron(config)#transmit-counter 4 port 18 only vlan 1 prio 7 enable
```

The above command creates and enables traffic counter 4 on port 18. The device will count the number of packets sent out on port 18 that are in VLAN 1 and have a priority queue of 7.

EXAMPLES:

To configure traffic counters for outbound traffic in a specific port region, enter a command such as the following:

```
FastIron(config)#transmit-counter 1 port 1 region vlan all prio all enable
```

The above command creates and enables traffic counter 1 on all ports that are in the same port region as port 1. The device will count the number of packets transmitted in this port region that belong to any VLAN and have any assigned priority queue.

Syntax: [no] transmit-counter <counter-ID> port [<slotnum>/]<port-num> only | region vlan <vlan-ID> | all priority <priority-queue> | all enable

Enter the **no** form of the command to remove the outbound traffic counter.

The <counter-ID> parameter identifies the traffic counter. You can configure up to 64 traffic counters. Enter a number from 1 – 64.

The <slotnum> parameter is required on chassis devices.

The <port-num> parameter is the port number to which enhanced traffic counters will apply. Enter the port number followed by **only** to apply the enhanced traffic counter to a specific port, or enter the port number followed by **region** to apply the enhanced traffic counter to all of the ports in the port region.

The <vlan-ID> parameter identifies the VLAN ID for which outbound traffic will be counted. Enter a number from 0 – 4095 or enter **all** to indicate all VLANs.

The <priority-queue> parameter identifies the 802.1p priority queue for which traffic will be counted. Enter a number from 0 – 7 or enter **all** to indicate all priority queues.

Displaying Enhanced Traffic Counter Profiles

To display the details of the traffic counters configured on your device, enter the **show transmit-counter profiles** command. The following shows an example output:

```
Router#show transmit-counter profiles

Tx Counter      Port(s)   Vlan Id  Priority  Device    Set
         1     1 -   12       All       All   Dev 0   Set0
         4           18         1         7   Dev 1   Set0
        10    13 -   24       100       All   Dev 1   Set1
```

Displaying Enhanced Traffic Counter Statistics

To display the traffic counters for outbound traffic, enter the **show transmit-counter profiles** command.

NOTE: Once the enhanced traffic counters are displayed, the counters are cleared (reset to zero).

The following shows an example output:

```
Router#show transmit-counter values 1

Transmit Queue Counter Values for Counter 1:
Transmitted Frames:
 Known Unicast               : 17204
 Multicast & Unknown Unicast : 2797
 Broadcast                   : 5
Dropped Frames:
 Bridge Egress Filtered      : 2
 Congestion Drops            : 0

Router#show transmit-counter values 4

Transmit Queue Counter Values for Counter 4:
Transmitted Frames:
 Known Unicast               : 124
 Multicast & Unknown Unicast : 2752
 Broadcast                   : 0
Dropped Frames:
 Bridge Egress Filtered      : 37
 Congestion Drops            : 0
```

Syntax: show transmit-counter values <number>

where <number> identifies a valid enhanced traffic counter and is a value from 1 – 64.

Table B.2: Outbound Traffic Counter Statistics

This Line...	Displays...
Transmitted Frames	
Known Unicast	The number of known unicast packets transmitted.
Multicast & Unknown Unicast	The number of multicast and unknown unicast packets transmitted.
Broadcast	The number of broadcast packets transmitted.
Dropped Frames	
Bridge Egress Filtered	The number of bridged outbound packets that were filtered and dropped. This number includes the number of packets that were dropped because of any one of the following conditions: • The port was disabled or the link was down. • The port or port region does not belong to the VLAN specified in the transmit counter configuration. • A Layer 2 protocol (e.g., spanning tree) had the port in a Blocked state. • The source port was suppressed for multi-target packets. • The priority queue specified in the traffic counter was not allowed for some other reason. • Unknown unicast and unregistered multicast packets were filtered.
Congestion Drops	The number of outbound packets that were dropped because of traffic congestion.

RMON Support

The Foundry RMON agent supports the following groups. The group numbers come from the RMON specification (RFC 1757).

- Statistics (RMON Group 1)
- History (RMON Group 2)
- Alarms (RMON Group 3)
- Events (RMON Group 9)

The CLI allows you to make configuration changes to the control data for these groups, but you need a separate RMON application to view and display the data graphically.

Maximum Number of Entries Allowed in the RMON Control Table

Starting in software release 03.0.01 for FastIron X Series devices, you can specify the maximum number of entries allowed in the RMON control table, including alarms, history, and events. In addition, the default number of RMON entries allowed in the RMON control table has increased from 240 to 1024 on the FESX, and 2048 on the FSX, FSX 800, and FSX 1600. The maximum number of RMON entries supported is 32768.

To set the maximum number of allowable entries to 3000 in the RMON history table, enter commands such as the following:

```
FastIron(config)#system-max rmon-entries 3000
FastIron(config)#write mem
FastIron(config)#exit
FastIron#reload
```

NOTE: You must save the change to the startup-config file and reload or reboot. The change does not take effect until you reload or reboot.

Syntax: system-max rmon-entries <value>

where <value> can be:

- 1536 – 32768 for FSX, FSX 800, and FSX 1600 devices
- 128 – 32768 for FESX devices

Statistics (RMON Group 1)

Count information on multicast and broadcast packets, total packets sent, undersized and oversized packets, CRC alignment errors, jabbers, collision, fragments and dropped events is collected for each port on a Foundry Layer 2 Switch or Layer 3 Switch.

No configuration is required to activate collection of statistics for the Layer 2 Switch or Layer 3 Switch. This activity is by default automatically activated at system start-up.

You can view a textual summary of the statistics for all ports by entering the following CLI command:

```
FastIron#show rmon statistics
Ethernet statistics 1 is active, owned by monitor
 Interface 1/1 (ifIndex 1) counters
                  Octets          0
             Drop events          0                 Packets          0
          Broadcast pkts          0          Multicast pkts          0
    CRC alignment errors          0          Undersize pkts          0
           Oversize pkts          0               Fragments          0
                 Jabbers          0              Collisions          0
          64 octets pkts          0     65 to 127 octets pkts          0
   128 to 255 octets pkts          0    256 to 511 octets pkts          0
  512 to 1023 octets pkts          0  1024 to 1518 octets pkts          0
```

Syntax: show rmon statistics [[<slotnum>/]<portnum>]

The <portnum> parameter specifies the port number. You can use the physical port number or the SNMP port number. The physical port number is based on the product. If you specify a physical port on a chassis device, you must also enter the slot number.

- If the product is a Compact device, the ports are numbered sequentially starting with 1.
- If the product is a Chassis device, the ports are numbered according to slot and port. For example, the first port in slot 1 is 1/1. The third port in slot 7 is 7/3.

The SNMP numbers of the ports start at 1 and increase sequentially. For example, if you are using a Chassis device and slot 1 contains an 8-port module, the SNMP number of the first port in slot 2 is 9. The physical port number of the same port is 2/1.

This command shows the following information.

Table B.3: Export Configuration and Statistics

This Line...	Displays...
Octets	The total number of octets of data received on the network. This number includes octets in bad packets. This number does not include framing bits but does include Frame Check Sequence (FCS) octets.
Drop events	Indicates an overrun at the port. The port logic could not receive the traffic at full line rate and had to drop some packets as a result. The counter indicates the total number of events in which packets were dropped by the RMON probe due to lack of resources. This number is not necessarily the number of packets dropped, but is the number of times an overrun condition has been detected.
Packets	The total number of packets received. This number includes bad packets, broadcast packets, and multicast packets.
Broadcast pkts	The total number of good packets received that were directed to the broadcast address. This number does not include multicast packets.
Multicast pkts	The total number of good packets received that were directed to a multicast address. This number does not include packets directed to the broadcast address.

Table B.3: Export Configuration and Statistics (Continued)

This Line...	Displays...
CRC alignment errors	The total number of packets received that were from 64 – 1518 octets long, but had either a bad FCS with an integral number of octets (FCS Error) or a bad FCS with a non-integral number of octets (Alignment Error). The packet length does not include framing bits but does include FCS octets.
Undersize pkts	The total number of packets received that were less than 64 octets long and were otherwise well formed. This number does not include framing bits but does include FCS octets.
Fragments	The total number of packets received that were less than 64 octets long and had either a bad FCS with an integral number of octets (FCS Error) or a bad FCS with a non-integral number of octets (Alignment Error). It is normal for this counter to increment, since it counts both runts (which are normal occurrences due to collisions) and noise hits. This number does not include framing bits but does include FCS octets.
Oversize packets	The total number of packets received that were longer than 1518 octets and were otherwise well formed. This number does not include framing bits but does include FCS octets.
Jabbers	The total number of packets received that were longer than 1518 octets and had either a bad FCS with an integral number of octets (FCS Error) or a bad FCS with a non-integral number of octets (Alignment Error). **Note**: This definition of jabber is different from the definition in IEEE-802.3 section 8.2.1.5 (10BASE5) and section 10.3.1.4 (10BASE2). These documents define jabber as the condition where any packet exceeds 20 ms. The allowed range to detect jabber is between 20 ms and 150 ms. This number does not include framing bits but does include FCS octets.
Collisions	The best estimate of the total number of collisions on this Ethernet segment.
64 octets pkts	The total number of packets received that were 64 octets long. This number includes bad packets. This number does not include framing bits but does include FCS octets.
65 to 127 octets pkts	The total number of packets received that were 65 – 127 octets long. This number includes bad packets. This number does not include framing bits but does include FCS octets.
128 to 255 octets pkts	The total number of packets received that were 128 – 255 octets long. This number includes bad packets. This number does not include framing bits but does include FCS octets.
256 to 511 octets pkts	The total number of packets received that were 256 – 511 octets long. This number includes bad packets. This number does not include framing bits but does include FCS octets.

Table B.3: Export Configuration and Statistics (Continued)

This Line...	Displays...
512 to 1023 octets pkts	The total number of packets received that were 512 – 1023 octets long. This number includes bad packets. This number does not include framing bits but does include FCS octets.
1024 to 1518 octets pkts	The total number of packets received that were 1024 – 1518 octets long. This number includes bad packets. This number does not include framing bits but does include FCS octets.

History (RMON Group 2)

All active ports by default will generate two history control data entries per active Foundry Layer 2 Switch port or Layer 3 Switch interface. An active port is defined as one with a link up. If the link goes down the two entries are automatically deleted.

Two history entries are generated for each device:

- A sampling of statistics every 30 seconds
- A sampling of statistics every 30 minutes

The history data can be accessed and displayed using any of the popular RMON applications

A sample RMON history command and its syntax is shown below:

```
FastIron(config)#rmon history 1 interface 1 buckets 10 interval 10 owner nyc02
```

Syntax: rmon history <entry-number> interface [<slotnum>/]<portnum> buckets <number> interval <sampling-interval> owner <text-string>

You can modify the sampling interval and the bucket (number of entries saved before overwrite) using the CLI. In the above example, owner refers to the RMON station that will request the information.

NOTE: To review the control data entry for each port or interface, enter the **show rmon history** command.

Alarm (RMON Group 3)

Alarm is designed to monitor configured thresholds for any SNMP integer, time tick, gauge or counter MIB object. Using the CLI, you can define what MIB objects are monitored, the type of thresholds that are monitored (falling, rising or both), the value of those thresholds, and the sample type (absolute or delta).

An alarm event is reported each time that a threshold is exceeded. The alarm entry also indicates the action (event) to be taken if the threshold be exceeded.

A sample CLI alarm entry and its syntax is shown below:

```
FastIron(config)#rmon alarm 1 ifInOctets.6 10 delta rising-threshold 100 1 falling
threshold 50 1 owner nyc02
```

Syntax: rmon alarm <entry-number> <MIB-object.interface-num> <sampling-time> <sample-type> <threshold-type> <threshold-value> <event-number> <threshold-type> <threshold-value> <event-number> owner <text-string>

Event (RMON Group 9)

There are two elements to the Event Group—the ***event control table*** and the ***event log table***.

The event control table defines the action to be taken when an alarm is reported. Defined events can be found by entering the CLI command, show event. The Event Log Table collects and stores reported events for retrieval by an RMON application.

A sample entry and syntax of the event control table is shown below:

```
FastIron(config)#rmon event 1 description ‘testing a longer string’ log-and-trap
public owner nyc02
```

Syntax: rmon event <event-entry> description <text-string> log | trap | log-and-trap owner <rmon-station>

sFlow

sFlow is a system for observing traffic flow patterns and quantities within and among a set of Layer 2 Switches and Layer 3 Switches. To support sFlow, participating Layer 2 and Layer 3 devices:

- Sample packet flows
- Collect the packet headers from sampled packets and collect ingress-egress information on these packets
- Compose the collected information into flow sample messages
- Relay these messages to an external device known as a collector

Participating devices also relay byte and packet counter data (counter samples) for ports to the collector.

sFlow is described in RFC 3176, “InMon Corporation's sFlow: A Method for Monitoring Traffic in Switched and Routed Networks”. Refer to this RFC to determine the contents of the sampled packet.

On the FastIron GS, you can use QoS queue 1 for priority traffic, even when sFlow is enabled on the port. This differs from the FastIron X Series. When sFlow is enabled on the FESX, FSX, and FWSX, these devices support seven priorities instead of eight because QoS queue 1 is reserved for sFlow and is not used by other packets. Any non-sFlow packets assigned to QoS queue 1 will be directed to QoS queue 0.

Configuration Considerations

This section lists the sFlow configuration considerations on Foundry’s FastIron devices.

Hardware Support

- FastIron devices support sFlow packet sampling of inbound traffic only. These devices do not sample outbound packets. However, FastIron devices support byte and packet count statistics for both traffic directions.
- sFlow is supported on all Ethernet ports (10/100, Gigabit, and 10 Gigabit)

Source Address

The sampled sFlow data sent to the collectors includes an agent_address field. This field identifies the IP address of the device that sent the data.

- On a Layer 2 Switch, agent_address is the Layer 2 Switch’s management IP address. You must configure the management IP address in order to export sFlow data from the device.
- On a Layer 3 Switch, sFlow looks for an IP address in the following order, and uses the first address found:
    - The router ID configured by the **ip router-id** command
    - The first IP address on the lowest-numbered loopback interface
    - The first IP address on the lowest-numbered virtual interface
    - The first IP address on any interface

NOTE: The device uses the router ID only if the device also has an IP interface with the same address.

NOTE: If an IP address in not already configured when you enable sFlow, the feature uses the source address 0.0.0.0. To display the agent_address, enable sFlow, then enter the **show sflow** command. See “Enabling sFlow Forwarding” on page B-17 and “Displaying sFlow Information” on page B-18.

NOTE: If you change the address sFlow will use for the agent_address, you must disable and re-enable sFlow to enable the feature to use the changed address.

Sampling Rate

The ***sampling rate*** is the average ratio of the number of packets incoming on an sFlow enabled port, to the number of flow samples taken from those packets. sFlow sampling can affect performance in some configurations.

Note that on the FastIron devices, the configured sampling rate and the actual rate are the same. The software does not adjust the configured sampling rate as on other Foundry devices.

Port Monitoring and sFlow

The FastIron GS supports sFlow and port monitoring together on the same ports.

On the FastIron GS, you can use QoS queue 1 for priority traffic, even when sFlow is enabled on the port. This differs from the FastIron X Series. When sFlow is enabled on the FESX, FSX, and FWSX, these devices support seven priorities instead of eight because QoS queue 1 is reserved for sFlow and is not used by other packets. Any non-sFlow packets assigned to QoS queue 1 will be directed to QoS queue 0.

- FESX and FWSX devices running software release 02.2.01 or later support port monitoring and sFlow together on the same device. The caveat is that these features cannot be configured together within the same port region. See “About Port Regions” on page 8-1 for a list of valid port regions.
- FSX devices running software release 02.2.00 or later support port monitoring and sFlow together on the same device. The caveat is that these features cannot be configured together within the same port region.

sFlow Support for IPv6 Packets

Foundry’s implementation of sFlow features provide support for IPv6 packets. This support includes extended router information and extended gateway information in the sampled packet. Note that sFlow support for IPv6 packets exists only on devices running software that supports IPv6.

Extended Router Information

Extended router information contains information for the next hop router. This information includes the next hop router’s IP address and the outgoing VLAN ID. Extended router information also includes the source IP address prefix length and the destination IP address prefix length.

Note that in IPv4, prefix length of source and destination IP addresses is collected only if BGP is configured on the devices. In IPv6, the information is collected if BGP is configured and once the route lookup is complete.

To obtain extended router information in IPv6 sampled packets, use “struct extended_router” as presented in RFC 3176.

Extended Gateway Information

Extended gateway information is included in an sFlow sampled packet if BGP is enabled. The extended gateway information includes the following BGP information about the packet’s destination route:

- This router’s autonomous system (AS) number
- The route’s source IP AS
- The route’s source peer AS
- The AS path to the destination

NOTE: AS communities and local preferences are not included in the sampled packets.

To obtain extended gateway information use “struct extended_gateway” as described in RFC 3176.

Configuring and Enabling sFlow

To configure sFlow:

- Specify collector information. The collector is the external device to which you are exporting the sFlow data. You can specify up to four collectors.
- Optional – Change the polling interval.
- Optional – Change the sampling rate.
- Enable sFlow globally.
- Enable sFlow forwarding on individual interfaces.

NOTE: If you change the router ID or other IP address value that sFlow uses for its agent_address, you need to disable and then re-enable sFlow to cause the feature to use the new source address.

Specifying the Collector

sFlow exports traffic statistics to an external collector. You can specify up to four collectors. You can specify more than one collector with the same IP address if the UDP port numbers are unique. You can have up to four unique combinations of IP address and UDP port number.

To specify sFlow collectors, enter a command such as the following:

```
FastIron(config)#sflow destination 10.10.10.1
```

This command specifies a collector with IP address 10.10.10.1, listening for sFlow data on UDP port 6343.

Syntax: [no] sflow destination <ip-addr> [<dest-udp-port>]

The <ip-addr> parameter specifies the collector’s IP address.

The <dest-udp-port> parameter specifies the UDP port on which the sFlow collector will be listening for exported sFlow data. The default port number is 6343.

The sampled sFlow data sent to the collectors includes an agent_address field. This field identifies the device that sent the data. See “Source Address” on page B-12.

Changing the Polling Interval

The polling interval defines how often sFlow byte and packet counter data for a port are sent to the sFlow collector(s). If multiple ports are enabled for sFlow, the Foundry device staggers transmission of the counter data to smooth performance. For example, if sFlow is enabled on two ports and the polling interval is 20 seconds, the Foundry device sends counter data every ten seconds. The counter data for one of the ports are sent after ten seconds, and counter data for the other port are sent after an additional ten seconds. Ten seconds later, new counter data for the first port are sent. Similarly, if sFlow is enabled on five ports and the polling interval is 20 seconds, the Foundry device sends counter data every four seconds.

The default polling interval is 20 seconds. You can change the interval to a value from 1 to any higher value. The interval value applies to all interfaces on which sFlow is enabled. If you set the polling interval to 0, counter data sampling is disabled.

To change the polling interval, enter a command such as the following at the global CONFIG level of the CLI:

```
FastIron(config)#sflow polling-interval 30
```

Syntax: [no] sflow polling-interval <secs>

The <secs> parameter specifies the interval and can be from 1 to any higher value. The default is 20 seconds. If you specify 0, counter data sampling is disabled.

Changing the Sampling Rate

The sampling rate is the average ratio of the number of packets incoming on an sFlow-enabled port, to the number of flow samples taken from those packets.

You can change the default (global) sampling rate. You also can change the rate on an individual port, overriding the default sampling rate of 512. With a sampling rate of 512, on average, one in every 512 packets forwarded on an interface is sampled.

Configuration Considerations

The sampling rate is a fraction in the form 1/N, meaning that, on average, one out of every N packets will be sampled. The **sflow sample** command at the global level or port level specifies N, the denominator of the fraction. Thus a higher number for the denominator means a lower sampling rate since fewer packets are sampled. Likewise, a lower number for the denominator means a higher sampling rate because more packets are sampled. For example, if you change the denominator from 512 to 128, the sampling rate increases because four times as many packets will be sampled.

NOTE: Foundry recommends that you do not change the denominator to a value lower than the default. Sampling requires CPU resources. Using a low denominator for the sampling rate can cause high CPU utilization.

Configured Rate and Actual Rate

When you enter a sampling rate value, this value is the ***configured rate***. The software rounds the value you enter to the next higher odd power of 2 to obtain the ***actual rate***. This value becomes the actual sampling rate. For example, if the configured sampling rate is 1000, then the actual rate is 2048 and 1 in 2048 packets are sampled by the hardware.

Change to Global Rate

If you change the global sampling rate, the change is applied to all sFlow-enabled ports ***except*** those ports on which you have already explicitly set the sampling rate. For example, suppose that sFlow is enabled on ports 1/1, 1/2, and 5/1. If you configure the sampling rate on port 1/1 but leave the other two ports using the default rate, then a change to the global sampling rate applies to ports 1/2 and 5/1 but not port 1/1. sFlow assumes that you want to continue using the sampling rate you explicitly configured on an individual port even if you globally change the sampling rate for the other ports.

Module Rate

While different ports on a module may be configured to have different sampling rates, the hardware for the module will be programmed to take samples at a single rate (the module sampling rate). The module sampling rate will be the highest sampling rate (i.e. lowest number) configured for any of the ports on the module.

When ports on a given module are configured with different sampling rates, the CPU discards some of the samples supplied by the hardware for ports with configured sampling rates which are lower than the module sampling rate. This is referred to as subsampling, and the ratio between the port sampling rate and the module sampling rate is known as the subsampling factor. For example, if the module in slot 4 has sFlow enabled on ports 4/2 and 4/8, and port 4/2 is using the default sampling rate of 512, and port 4/8 is configured explicitly for a rate of 2048, then the module sampling rate will be 512 because this is this highest port sampling rate (lowest number). The subsampling factor for port 4/2 will be 1, meaning that every sample taken by the hardware will be exported, while the subsampling factor for port 4/8 will be 4, meaning that one out of every four samples taken by the hardware will be exported. Whether a port's sampling rate is configured explicitly, or whether it uses the global default setting, has no effect on the calculations.

You do not need to perform any of these calculations to change a sampling rate. For simplicity, the syntax information in this section lists the valid sampling rates. In addition, the software will round the value you enter up to the nearest value listed. You can display the rates you entered (the configured rates) as well as the rates rounded up to by the software (the actual rates) for the default sampling rate, module rates, and all sFlow-enabled ports by entering the **show sflow** command. See “Displaying sFlow Information” on page B-18.

Sampling Rate for New Ports

When you enable sFlow on a port, the port's sampling rate is set to the global default sampling rate. This also applies to ports on which you disable and then re-enable sFlow. The port does not retain the sampling rate it had when you disabled sFlow on the port, even if you had explicitly set the sampling rate on the port.

Changing the Default Sampling Rate

To change the default (global) sampling rate, enter a command such as the following at the global CONFIG level of the CLI:

```
FastIron(config)#sflow sample 2048
```

Syntax: [no] sflow sample <num>

The <num> parameter specifies the average number of packets from which each sample will be taken. The software rounds the value you enter to the next higher odd power of 2. This value becomes the actual default sampling rate and is one of the following.

- 2
- 8
- 32
- 128
- 512
- 2048
- 4096
- 8192
- 32768
- 131072
- 524288
- 2097152
- 8388608
- 33554432
- 134217728
- 536870912
- 2147483648

For example, if the configured sampling rate is 1000, then the actual rate is 2048 and 1 in 2048 packets are sampled by the hardware.

Changing the Sampling Rate of a Module

You cannot change a module's sampling rate directly. You can change a module's sampling rate only by changing the sampling rate of a port on that module.

Changing the Sampling Rate on a Port

You can configure an individual port to use a different sampling rate than the global default sampling rate. This is useful in cases where ports have different bandwidths. For example, if you are using sFlow on 10/100 ports and Gigabit Ethernet ports, you might want to configure the Gigabit ports to use a higher sampling rate (and thus gather fewer samples per number of packets) than the 10/100 ports.

To change the sampling rate on an individual port, enter a command such as the following at the configuration level for the port:

```
FastIron(config-if-1/1)#sflow sample 8192
```

Syntax: [no] sflow sample <num>

The <num> parameter specifies the average number of packets from which each sample will be taken. The software rounds the value you enter up to the next odd power of 2. The actual sampling rate becomes one of the values listed in “Changing the Default Sampling Rate” .

Enabling sFlow Forwarding

sFlow exports data only for the interfaces on which you enable sFlow forwarding. You can enable sFlow forwarding on Ethernet interfaces.

To enable sFlow forwarding:

- Globally enable the sFlow feature.
- Enable sFlow forwarding on individual interfaces.

NOTE: Before you enable sFlow, make sure the device has an IP address that sFlow can use as its source address. See “Source Address” on page B-12 for the source address requirements.

NOTE: When you enable sFlow forwarding on an 802.1X-enabled interface, the samples taken from the interface include the username used to obtain access to the inbound and/or outbound ports, if that information is available. For information about 802.1X, see the chapter “Configuring 802.1X Port Security” on page 42-1.

Enabling sFlow Forwarding

To enable sFlow forwarding, enter commands such as the following:

```
FastIron(config)#sflow enable
FastIron(config)#interface ethernet 1/1 to 1/8
FastIron(config-mif-1/1-1/8)#sflow forwarding
```

These commands globally enable sFlow, then enable sFlow forwarding on Ethernet ports 1/1 – 1/8. You must use both the **sflow enable** and **sflow forwarding** commands to enable the feature.

Syntax: [no] sflow enable

Syntax: [no] sflow forwarding

Displaying sFlow Information

To display sFlow configuration information and statistics, enter the following command at any level of the CLI:

```
FastIron#show sflow
sFlow services are enabled.
sFlow agent IP address: 123.123.123.1
4 collector destinations configured:
Collector IP 192.168.4.204, UDP 6343
Collector IP 192.168.4.200, UDP 6333
Collector IP 192.168.4.202, UDP 6355
Collector IP 192.168.4.203, UDP 6565
Polling interval is 0 seconds.
Configured default sampling rate: 1 per 512 packets.
Actual default sampling rate: 1 per 512 packets.
10552 UDP packets exported
24127 sFlow samples collected.
sFlow ports: ethe 1/2 to 1/12 ethe 1/15 ethe 1/25 to 1/26 ethe 4/1 ethe 5/10 to
5/20 ethe 8/1 ethe 8/4
Module Sampling Rates
---------------------
Slot  1 configured rate=512, actual rate=512
Slot  3 configured rate=0, actual rate=0
Slot  4 configured rate=10000, actual rate=32768
Slot  5 configured rate=512, actual rate=512
Slot  7 configured rate=0, actual rate=0
Slot  8 configured rate=512, actual rate=512
Port Sampling Rates
-------------------
Port 8/4, configured rate=512, actual rate=512, Subsampling factor=1
Port 8/1, configured rate=512, actual rate=512, Subsampling factor=1
Port 5/20, configured rate=3000, actual rate=8192, Subsampling factor=16
Port 5/19, configured rate=512, actual rate=512, Subsampling factor=1
Port 5/18, configured rate=512, actual rate=512, Subsampling factor=1
Port 5/17, configured rate=1500, actual rate=2048, Subsampling factor=4
Port 5/16, configured rate=1500, actual rate=2048, Subsampling factor=4
Port 5/15, configured rate=1500, actual rate=2048, Subsampling factor=4
Port 5/14, configured rate=1500, actual rate=2048, Subsampling factor=4
Port 5/13, configured rate=512, actual rate=512, Subsampling factor=1
Port 5/12, configured rate=512, actual rate=512, Subsampling factor=1
Port 5/11, configured rate=512, actual rate=512, Subsampling factor=1
Port 5/10, configured rate=512, actual rate=512, Subsampling factor=1
Port 4/1, configured rate=10000, actual rate=32768, Subsampling factor=1
Port 1/26, configured rate=512, actual rate=512, Subsampling factor=1
Port 1/25, configured rate=512, actual rate=512, Subsampling factor=1
Port 1/15, configured rate=512, actual rate=512, Subsampling factor=1
Port 1/12, configured rate=512, actual rate=512, Subsampling factor=1
Port 1/11, configured rate=512, actual rate=512, Subsampling factor=1
Port 1/10, configured rate=512, actual rate=512, Subsampling factor=1
Port 1/9, configured rate=512, actual rate=512, Subsampling factor=1
Port 1/8, configured rate=512, actual rate=512, Subsampling factor=1
Port 1/7, configured rate=1000, actual rate=2048, Subsampling factor=4
Port 1/6, configured rate=512, actual rate=512, Subsampling factor=1
Port 1/5, configured rate=512, actual rate=512, Subsampling factor=1
Port 1/4, configured rate=512, actual rate=512, Subsampling factor=1
Port 1/3, configured rate=512, actual rate=512, Subsampling factor=1
Port 1/2, configured rate=1000, actual rate=2048, Subsampling factor=4
```

Syntax: show sflow

This command shows the following information.

Table B.4: sFlow Information

This Field...	Displays...
sFlow services	The feature state, which can be one of the following: • disabled • enabled
sFlow agent IP address	The IP address that sFlow is using in the agent_address field of packets sent to the collectors. See “Source Address” on page B-12.
Collector	The collector information. The following information is displayed for each collector: • IP address • UDP port If more than one collector is configured, the line above the collectors indicates how many have been configured.
Polling interval	The port counter polling interval.
Configured default sampling rate	The configured global sampling rate. If you changed the global sampling rate, the value you entered is shown here. The actual rate calculated by the software based on the value you entered is listed on the next line, ”Actual default sampling rate”.
Actual default sampling rate	The actual default sampling rate.
UDP packets exported	The number of sFlow export packets the Foundry device has sent. **Note**: Each UDP packet can contain multiple samples.
sFlow samples collected	The number of sampled packets that have been sent to the collector(s).
sFlow ports	The ports on which you enabled sFlow.
Module Sampling Rates	The configured and actual sampling rates for each module. If a module does not have any sFlow-enabled ports, the rates are listed as 0.
Port Sampling Rates	The configured and actual sampling rates for each sFlow-enabled port. The Subsampling factor indicates how many times the sampling rate of the port's module is multiplied to achieve the port's sampling rate. Because of the way the actual sampling rates are computed, the Subsampling factors are always whole numbers.

Clearing sFlow Statistics

To clear the UDP packet and sFlow sample counters in the **show sflow** display, enter the following command:

```
FastIron#clear statistics
```

Syntax: clear statistics

This command clears the values in the following fields of the **show sflow** display:

- UDP packets exported
- sFlow samples collected

NOTE: This command also clears the statistics counters used by other features.

Configuring a Utilization List for an Uplink Port

You can configure uplink utilization lists that display the percentage of a given uplink port's bandwidth that is used by a specific list of downlink ports. The percentages are based on 30-second intervals of RMON packet statistics for the ports. Both transmit and receive traffic is counted in each percentage.

NOTE: This feature is intended for ISP or collocation environments in which downlink ports are dedicated to various customers' traffic and are isolated from one another. If traffic regularly passes between the downlink ports, the information displayed by the utilization lists does not provide a clear depiction of traffic exchanged by the downlink ports and the uplink port.

Each uplink utilization list consists of the following:

- Utilization list number (1, 2, 3, or 4)
- One or more uplink ports
- One or more downlink ports

Each list displays the uplink port and the percentage of that port's bandwidth that was utilized by the downlink ports over the most recent 30-second interval.

You can configure up to four bandwidth utilization lists.

Command Syntax

To configure an uplink utilization list, enter commands such as the following. The commands in this example configure a link utilization list with port 1/1 as the uplink port and ports 1/2 and 1/3 as the downlink ports.

```
FastIron(config)#relative-utilization 1 uplink eth 1/1 downlink eth 1/2 to 1/3
FastIron(config)#write memory
```

Syntax: [no] relative-utilization <num> uplink ethernet [<slotnum>/]<portnum> [to [<slotnum>/]<portnum> | [<slotnum>/]<portnum>…] downlink ethernet [<slotnum>/]<portnum> [to [<slotnum>/]<portnum> | [<slotnum>/]<portnum>…]

The <num> parameter specifies the list number. You can configure up to four lists. Specify a number from 1 – 4.

The **uplink ethernet** parameters and the port numbers you specify after the parameters indicate the uplink ports.

The **downlink ethernet** parameters and the port numbers you specify after the parameters indicate the downlink ports.

Displaying Utilization Percentages for an Uplink

After you configure an uplink utilization list, you can display the list to observe the percentage of the uplink's bandwidth that each of the downlink ports used during the most recent 30-second port statistics interval. The number of packets sent and received between the two ports is listed, as well as the ratio of each individual downlink port's packets relative to the total number of packets on the uplink.

To display an uplink utilization list, enter a command such as the following at any level of the CLI:

```
FastIron#show relative-utilization 1
uplink: ethe 1
30-sec total uplink packet count = 3011
packet count ratio (%)
```

```
  1/ 2:60   1/ 3:40
```

In this example, ports 1/2 and 1/3 are sending traffic to port 1/1. Port 1/2 and port 1/3 are isolated (not shared by multiple clients) and typically do not exchange traffic with other ports except for the uplink port, 1/1.

Syntax: show relative-utilization <num>

The <num> parameter specifies the list number.

NOTE: The example above represents a pure configuration in which traffic is exchanged only by ports 1/2 and 1/1, and by ports 1/3 and 1/1. For this reason, the percentages for the two downlink ports equal 100%. In some cases, the percentages do not always equal 100%. This is true in cases where the ports exchange some traffic with other ports in the system or when the downlink ports are configured together in a port-based VLAN.

In the following example, ports 1/2 and 1/3 are in the same port-based VLAN.

```
FastIron#show relative-utilization 1
uplink: ethe 1
30-sec total uplink packet count = 3011
packet count ratio (%)
  1/ 2:100   1/ 3:100
```

Here is another example showing different data for the same link utilization list. In this example, port 1/2 is connected to a hub and is sending traffic to port 1/1. Port 1/3 is unconnected.

```
FastIron#show relative-utilization 1
uplink: ethe 1
30-sec total uplink packet count = 2996
packet count ratio (%)
  1 /2:100   1/ 3:---
```

Appendix C
Software Specifications

IEEE Compliance

Foundry devices support the following standards:

Table 47.3: IEEE Compliance

Standard	Description	FESX, FSX, FSX 800, FSX 1600, FWSX[1]	FGS, FLS
802.1AB	Station and Media Access Control Connectivity Discovery Also supports TIA-1057, Telecommunications – IP Telephony Infrastructure -– Link Layer Discovery Protocol (LLDP) for Media Endpoint Devices	X	X
802.1d	Bridging	X	X
802.1D	1998	X	X
802.1p/q	VLAN Tagging and Priority	X	X
802.1w	Rapid Spanning Tree	X	X
802.1x	Port-based Authentication, Dynamic VLAN, ACL, and MAC Filter Group Assignment	X	X
802.3	10Base-T	X	X
802.3	Ethernet Like MIB	X	X
802.3ab	1000Base-T		X
802.3ad	Link Aggregation (Dynamic and Static) and Trunk Groups	X	X
802.3ae	10 Gigabit Ethernet	X	X

Table 47.3: IEEE Compliance (Continued)

802.3af	Power over Ethernet	X	X
802.3u	100Base-TX, 1000Base-FX	X	X
802.3z	1000Base-SX, 1000Base-LX, 1000Base-T	X	X
802.3x	Flow Control	X	X
	Ethernet Interface MIB	X	X
	PVSTP/+	X	X
	Repeater MIB	X	X
	SNMP MIB II	X	X
	SNMP V1, V2c, and V3	X	X

1.The FWSX does not support Layer 3 features.

RFC Support

The following table lists the RFCs supported by Foundry devices.

NOTE: Some devices support only a subset of the RFCs. For example, Layer 2 Switches do not support router-specific RFCs. For a list of features supported on your device, see the data sheet or the software release notes for the version of software running on your device.

Table 47.4: Foundry RFC Support

RFC Number	Protocol or Standard	FESX, FSX, FSX 800, FSX 1600, FWSX[1]	FGS and FLS
768	User Datagram Protocol (UDP)	X	X
783	Trivial File Transfer Protocol (TFTP)	X	X
791	Internet Protocol (IP)	X	X
792	Internet Control Message Protocol (ICMP)	X	X
793	Transmission Control Protocol (TCP)	X	X
826	Ethernet Address Resolution Protocol (ARP)	X	X
854, 855, and 857	Telnet	X	X
894	IP over Ethernet frames	X	X
903	Reverse ARP (RARP)	X	
906	Bootstrap loading using TFTP	X	
919	Broadcast Internet datagrams	X	
920	Domain requirements	X	
922	Broadcast Internet datagrams in the presence of subnets	X	
950	Internet standard subnetting procedure	X	
951	Bootstrap Protocol (BootP)	X	
1027	Proxy ARP	X	
1042	IP datagrams over IEEE 802 networks (for Ethernet)	X	X
1058	Route Information Protocol (RIP) version 1	X	
1075	Distance Vector Multicast Routing Protocol	X	
1112	Internet Gateway Management Protocol (IGMP) version 1	X	
1122 and 1123	Requirements for Internet hosts (routers)	X	

Table 47.4: Foundry RFC Support (Continued)

RFC Number	Protocol or Standard	FESX, FSX, FSX 800, FSX 1600, FWSX[1]	FGS and FLS
1141	Incremental updating of the Internet checksum	X	
1155	Structure and Identification of Management Information (SMI)	X	X
1157	Simple Network Management Protocol (SNMP) version 1	X	X
1191	Path MTU Discovery	X	
1212	Concise MIB Definitions	X	
1213	MIB II Definitions	X	X
1215	SNMP generic traps	X	X
1256	ICMP Router Discovery Protocol (IRDP)	X	
1267	Border Gateway Protocol version 3	X	
1269	Definitions of Managed Objects for the Border Gateway Protocol: Version 3	X	
1321	The MD5 Message-Digest Algorithm	X	X
1340	Assigned numbers (where applicable)	X	
1354	IP Forwarding Table MIB	X	
1377	The PPP OSI Network Layer Control Protocol (OSINLCP)	X	
1398	Ethernet-Like MIB	X	X
1492	An Access Control Protocol, Sometimes Called TACACS	X	X
1493	Bridge MIB (excluding filtering of objects)	X	X
1516	Repeater MIB	X	X
1519	Classless Inter-Domain Routing (CIDR): an Address Assignment and Aggregation Strategy	X	X
1541 and 1542	Dynamic Host Configuration Protocol (DHCP)	X	
1573	SNMP MIB II	X	X
1583	Open Shortest Path First (OSPF)	X	
1587	OSPF Not-So-Stubby Areas (NSSAs)	X	
1591	Domain Name System Structure and Delegation	X	
1643	Ethernet Interface MIB	X	X

Table 47.4: Foundry RFC Support (Continued)

RFC Number	Protocol or Standard	FESX, FSX, FSX 800, FSX 1600, FWSX[1]	FGS and FLS
1657	Definitions of Managed Objects for the Fourth Version of the Border Gateway Protocol (BGP4) using SMIv2	X	
1661	The Point-to-Point Protocol (PPP)	X	
1723	RIP version 2	X	
1724	RIP version 2 MIB	X	
1742	AppleTalk Management Information Base II	X	
1745	OSPF Interactions	X	
1757	Remote Monitoring (RMON) groups 1, 2, 3, 9	X	
1765	OSPF Database Overflow	X	
1771	Border Gateway Protocol version 4 (BGP4)	X	
1812	Requirements for IP version 4 routers	X	
1850	OSPF version 2 MIB	X	
1905	Protocol Operations for version 2 of the Simple Network Management Protocol (SNMPv2)	X	X
1906	Transport Mappings for version 2 of the Simple Network Management Protocol (SNMPv2)	X	X
1965	Autonomous System Configurations for BGP4	X	
1966	BGP Route Reflection	X	
1997	BGP Communities Attributes	X	
2003	IP Tunnelling	X	
2011	SNMPv2 Management Information Base for the Internet Protocol using SMIv2	X	
2012	SNMPv2 Management Information Base for the Transmission Control Protocol using SMIv2	X	
2013	SNMPv2 Management Information Base for the User Datagram Protocol using SMIv2	X	
2068	HTTP	X	
2096	IP Forwarding MIB	X	
2131	BootP/DHCP Relay	X	
2138	Remote Authentication Dial In User Server (RADIUS)	X	X
2139	RADIUS Accounting	X	
2154	OSPF with Digital Signatures (Password, MD-5)	X	

Table 47.4: Foundry RFC Support (Continued)

RFC Number	Protocol or Standard	FESX, FSX, FSX 800, FSX 1600, FWSX[1]	FGS and FLS
2178	Open Shortest Path First (OSPF)	X	
2205	Resource ReSerVation Protocol (RSVP) -- version 1 Functional Specification	X	
2233	The Interfaces Group MIB using SMIv2	X	
2236	Internet Gateway Management Protocol (IGMP) version 2	X	
2239	802.3 Medium Attachment Units (MAUs) using SMIv2	X	X
2283	Multiprotocol Extensions for BGP4	X	
2328	OSPF version 2 **Note**: AS External LSA reduction is supported.	X	
2336	IGMP version 2	X	
2338	Virtual Router Redundancy Protocol (VRRP)	X	
2362	IP Multicast PIM Sparse	X	
2370	The OSPF Opaque LSA Option	X	
2385	TCP MD5 Signature Option (for BGP4)	X	
2439	BGP Route Flap Dampening	X	
2482	Language Tagging in Unicode Plain Text	X	
2544	Benchmarking Methodology for Network Interconnect Devices	X	X
2570	Introduction to version 3 of the Internet-standard Network Management Framework	X	X
2571	An Architecture of Describing SNMP Management Frameworks	X	X
2572	Message Processing and Dispatching for the Simple Network Management Protocol (SNMP)	X	X
2573	SNMP version 3 Applications	X	X
2574	User-based Security (USM) for version 3 of the Simple Network Management Protocol (SNMPv3)	X	X
2575	View-based Access Control Model (VACM) for the Simple Network Management Protocol (SNMP)	X	X
2576	Coexistence between Version 1, Version 2, and Version 3 of the Internet-standard Network Management Framework	X	

Table 47.4: Foundry RFC Support (Continued)

RFC Number	Protocol or Standard	FESX, FSX, FSX 800, FSX 1600, FWSX[1]	FGS and FLS
2578	Structure of Management Information Version 2 (SMIv2)	X	X
2579	Textual Conventions for SMIv2	X	
2580	Conformance Statements for SMIv2	X	
2665	Ethernet Like MIB (incorporates RFC 1398)	X	X
2674	Definitions of Managed Objects for Bridges with Traffic Classes, Multicast Filtering and Virtual LAN Extensions	X	
2787	Definitions of Managed Objects for the Virtual Router Redundancy Protocol	X	
2796	BGP Route Reflection	X	
2842	BGP Capability Advertisement	X	
2865	Remote Authentication Dial In User Service (RADIUS)	X	X
2866	RADIUS Accounting	X	
2869	RADIUS Extensions	X	
2889	Benchmarking Methodology for LAN Switching Devices	X	X
2918	Route Refresh Capability for BGP4	X	
2932	IPv4 Multicast Routing MIB	X	
2933	Internet Group Management Protocol MIB	X	
2934	Protocol Independent Multicast MIB for IPv4	X	
3176	InMon Corporation's sFlow: A Method for Monitoring Traffic in Switched and Routed Networks	X	
3411	Simple Network Management Protocol (SNMP) Management Frameworks	X	X
3412	Message Processing and Dispatching for the Simple Network Management Protocol (SNMP)	X	X
3413	Simple Network Management Protocol (SNMP) Applications	X	X
3414	User-based Security Model (USM) for version 3 of the Simple Network Management Protocol (SNMPv3)	X	X
3415	View-based Access Control Model (VACM) for the Simple Network Management Protocol (SNMP)	X	X

Table 47.4: Foundry RFC Support (Continued)

RFC Number	Protocol or Standard	FESX, FSX, FSX 800, FSX 1600, FWSX[1]	FGS and FLS
3416	Version 2 of the Protocol Operations for the SNMP	X	X
3418	Management Information Base (MIB) for the Simple Network Management Protocol (SNMP)	X	X
3584	Coexistence between Version 1, Version 2, and Version 3 of the Internet-standard Network Management Framework	X	X
3918	Benchmarking Methodology for IP Multicast	X	X
4251	The Secure Shell (SSH) Protocol Architecture	X	X
4252	The Secure Shell (SSH) Authentication Protocol	X	X
4253	The Secure Shell (SSH) Transport Protocol	X	X
4254	The Secure Shell (SSH) Connection Protocol	X	X
4330	Simple Network Time Protocol (SNTP) version 4	X	
	AAA	X	
	Bi-level access mode (standard and EXEC level)	X	
	DVMRP V3-07	X	
	HTTP and HTTPS	X	
	IGMP Snooping (versions 1, 2, and 3)	X	X
	IGMP version 3	X	
	Integrated standard-based Command Line Interface (CLI)	X	X
	IronView Network Manager (INM) web-based graphical user interface	X	X
	MRP	X	X
	PIM-DM V1	X	
	PIM-SSM	X	
	Protection for Denial of Service attacks, such as TCP SYN or Smurf Attacks	X	X
	RMON HP OpenView for Sun Solaris, HP-UX, IBM’s AIX, and Windows NT	X	X
	Secure Copy (SCP)	X	X
	SSH V 1.5	X	X
	SSH V 2	X	
	TACACS/TACACS+	X	X

Table 47.4: Foundry RFC Support (Continued)

RFC Number	Protocol or Standard	FESX, FSX, FSX 800, FSX 1600, FWSX[1]	FGS and FLS
	TELNET and SSH V1	X	X
	UDLD	X	X
	Username/Password (challenge and response)	X	X

[1] The FWSX does not support Layer 3 features.

Internet Drafts

In addition to the RFCs listed in “RFC Support” on page C-3, the Layer 3 Switches support the following Internet drafts:

- ietf-idmr-dvmrp version 3.05, obsoletes RFC 1075
- draft-ietf-magma-igmp-proxy.txt
- draft-ietf-pim-dm-05 (V1)
- draft-ietf-pim-v2-dm-03 (V2)
- draft-katz-yeung-ospf-traffic-03.txt
- TACACS+ Protocol version 1.78

NOTE: Foundry supports the portions of this draft that describe the Extended IP reachablity TLV (TLV type 135) and the extended Intermediate System (IS) reachability TLV (TLV type 22) to provide support for wide metrics.