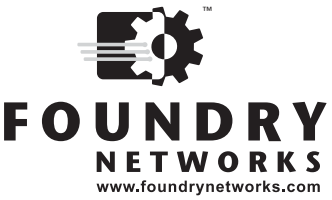
Foundry Enterprise Configuration and Management Guide



2100 Gold Street
P.O. Box 649100
San Jose, CA 95164-9100
Tel 408.586.1700
Fax 408.586.1900

January 2006

Contents

CHAPTER 12
CONFIGURING IP 12-1

APPENDIX B
POLICIES AND FILTERS B-1

Chapter 1
Getting Started

This guide describes the advanced configuration features in the following devices:

- Enterprise IronWare software releases, which apply to the following products:
  - NetIron 400/800/1500 Chassis devices with IronCore or JetCore management modules
  - BigIron 4000/8000/15000 Chassis devices with IronCore or JetCore management modules
  - FastIron II, FastIron II Plus, and FastIron III with M2 or higher management modules
  - FastIron 400/800/1500 Chassis devices with JetCore modules
  - FastIron 4802 Stackable device
- Service Provider IronWare software releases, which apply to the following products:
  - NetIron 400/800/1500 Chassis devices with IronCore or JetCore management modules
  - BigIron 4000/8000/15000 Chassis devices with IronCore or JetCore management modules
  - NetIron 4802 Stackable device
  - FastIron 4802 Stackable device

  NOTE: You cannot use this software on FastIron Chassis devices.
- Terathon devices that include the following:
  - BigIron MG8
  - NetIron 40G
  - NetIron IMR 640
- FastIron Edge Switch
- IronPoint-FastIron Edge Switch (IP-FES) Release 01.3.00 through 01.4.01
- ServerIron product family

For a list of enhancements in this edition, see the *Foundry Switch and Router Installation and Basic Configuration Guide*.

Audience

This manual is designed for system administrators with a working knowledge of Layer 2 and Layer 3 switching and routing.

If you are using a Foundry Layer 3 Switch, you should be familiar with the following protocols if applicable to your network – IP, RIP, OSPF, IS-IS, BGP4, MBGP, IGMP, PIM, DVMRP, IPX, AppleTalk, FSRP, 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 Switch and Router Installation and Basic Configuration Guide* – provides configuration guidelines for Layer 2 and Layer 3 devices and installation procedures for the Foundry devices with IronCore and JetCore modules.
- *Foundry Security Guide* – provides procedures for securing management access to Foundry devices and for protecting against Denial of Service (DoS) attacks.
- *Foundry Enterprise Configuration and Management Guide* – provides configuration information for enterprise routing protocols including IP, RIP, IP multicast, OSPF, BGP4, VRRP and VRRPE.
- *Foundry NetIron Service Provider Configuration and Management Guide* – provides configuration information for IS-IS and MPLS for Foundry devices that support IS-IS and MPLS, except for the NetIron IMR 640.
- *Foundry NetIron IMR 640 Service Provider Configuration and Management Guide* – provides configuration information for IS-IS and MPLS for for the NetIron IMR 640.
- *Foundry Switch and Router Command Line Interface Reference* – provides a list and syntax information for all the Layer 2 Switch and Layer 3 Switch CLI commands.
- *Foundry Diagnostic Guide* – provides descriptions of diagnostic commands that can help you diagnose and solve issues on Layer 2 Switches and Layer 3 Switches.
- *Foundry BigIron Mg8 Switch Installation and Basic Configuration Guide* – provides installation procedures for the BigIron MG8. This guide also presents the management modules available in the device.
- *Foundry NetIron 40G Switch Installation and Basic Configuration Guide* – provides installation procedures for the BigIron MG8. This guide also presents the management modules available in the device.
- *NetIron IMR 640 Installation and Basic Configuration Guide* – provides procedures for installing modules into and connecting your DC power source(s) to the NetIron IMR 640 chassis, cabling the Ethernet interface ports, and performing a basic configuration of the software.
- *Foundry Management Information Base Reference* – presents the Simple Network Management Protocol (SNMP) Management Information Base (MIB) objects that are supported in the Foundry devices.
- *Foundry IPv6 Configuration Guide* – provide configuration information for IPv6 features.
- *Foundry IronPoint Wireless LAN Configuration Guide* – presents the features for the IronPoint wireless LAN (WLAN).

To order additional copies of these manuals, do one of the following:

- Call 1.877.TURBOCALL (887.2622) in the United States or 1.408.586.1881 outside the United States.
- Send email to info@foundrynet.com.

How to Get Help

Foundry Networks technical support will ensure that the fast and easy access that you have come to expect from your Foundry Networks products will be maintained.

Web Access

- http://www.foundrynetworks.com

Email Access

Technical requests can also be sent to the following email address:

- support@foundrynet.com

Telephone Access

- 1.877.TURBOCALL (887.2622) United States
- 1.408.586.1881 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
Configuring Basic Quality of Service

This chapter describes how to configure Quality of Service (QoS) on Foundry devices.

NOTE: To configure QoS on a FastIron Edge Switch, see "Configuring Quality of Service on a FastIron Edge Switch and FastIron Edge Switch X-Series" on page 3-1

You can configure the following QoS features on Foundry devices:

- Choose between a strict queuing method and a weighted queuing method.
- Modify the minimum guaranteed percentage of bandwidth for each queue.
- Apply a QoS profile (one of the four queues) to 802.1q tagged VLAN packets.

NOTE: The QoS features listed above apply only to Chassis devices, the FastIron 4802, and the TurboIron/8. The following features, except IP Type of Service (ToS)-based QoS, apply to all products.

- Assign QoS priorities to traffic.
- Display the percentage of an uplink's bandwidth that each of a given set of ports uses. This is especially useful in environments where collocated customers on different, isolated ports share common uplink ports.
- Configure IP Type of Service (ToS)-based QoS.

NOTE: The ToS-based QoS described in this chapter applies only to the NetIron stackable Layer 3 Switch. To configure ToS-based QoS on a JetCore device, VM1 module, or 10 Gigabit Ethernet module, see "Configuring Enhanced Quality of Service" on page 4-1. To configure ToS-based QoS on a FastIron Edge Switch, see "Configuring Quality of Service on a FastIron Edge Switch and FastIron Edge Switch X-Series" on page 3-1

The Queues

Chassis devices, the FastIron 4802, and the TurboIron/8 use the following queues:

- qosp3 – The highest priority queue. This queue corresponds to 802.1p prioritization levels 6 and 7 and Foundry priority levels 6 and 7.
- qosp2 – The second-highest priority queue. This queue corresponds to 802.1p prioritization levels 4 and 5 and Foundry priority levels 4 and 5.
- qosp1 – The third-highest priority queue. This queue corresponds to 802.1p prioritization levels 2 and 3 and

Foundry priority levels 2 and 3.

- qosp0 – The lowest priority queue. This queue corresponds to 802.1p prioritization levels 0 and 1 and Foundry priority levels 0 and 1.

The queue names listed above are the default names. You can rename the queues on Chassis devices, the FastIron 4802, and the TurboIron/8 if you want, as described in "Renaming the Queues" on page 2-4.

Stackable devices (other than the FastIron 4802 and TurboIron/8) have two queues:

- High
- Normal

All traffic is classified in the normal queue by default. The devices forward all high priority traffic on a port's outbound queue before forwarding normal priority traffic on the port.

You can classify packets and assign them to specific queues based on the following criteria:

- Incoming port (also called ingress port)
- IP source and destination addresses
- Layer 4 source and destination information (for all IP addresses or specific IP addresses)
- Static MAC entry
- AppleTalk socket number
- Layer 2 port-based VLAN membership
- 802.1q tag

By default, all the traffic types listed above except the 802.1q tagged packets are in the best effort queue, which is the lowest priority queue. The 802.1q tagged packets are assigned to a queue based on the priority level (0 – 7) in the packet's tag. The default mapping of the priority levels to the queues is as follows.

Priority Level	Queue
6, 7	qosp3
4, 5	qosp2
2, 3	qosp1
0, 1	qosp0

In cases where a packet matches more than one traffic type, the highest queue level among the traffic type is used. For example, if a tagged packet arrives on a tagged port and the 802.1p priority is 4 (qosp2) but the packet contains IP source and destination information that matches an IP access policy configured to assign the traffic to priority 7 (qosp3), the device places the packet in qosp3 of the outbound port.

Automatic Queue Mapping for IP Type Of Service (TOS) Values

Foundry devices that support IronClad QoS automatically examine the first two bits in the Type of Service (TOS) header in each IP packet as it enters the device on a 10/100 port. The device then places the packet in the QoS queue that corresponds to the TOS value.

The TOS value in the first two bits can be one of the following.

TOS value (binary)	Queue
11	qosp3

TOS value (binary)	Queue
10	qosp2
01	qosp1
00	qosp0

As the packet moves through the system, if the packet matches other QoS allocations you have configured, the packet is moved into a higher queue accordingly. For example, if the TOS values place the packet in qosp1, but the packet is part of a port-based VLAN that is in qosp3, the packet enters queue qosp3. Packets can enter higher queues but never enter lower queues as they move through the system.

NOTE: The TOS mapping applies only to IP packets received on 10/100 ports. It does not apply to Gigabit or POS ports.

Queuing Methods

You can configure a Chassis device, the FastIron 4802, or the TurboIron/8 to use one of the following queuing methods:

- Weighted – A weighted fair queuing algorithm is used to rotate service among the four queues. The rotation is based on the weights you assign to each queue. This is the default queuing method and uses a default set of queue weights. This method rotates service among the four queues, forwarding a specific number of packets in one queue before moving on to the next one.

  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.

- Strict – 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.

Selecting the Queuing Method

Foundry Chassis devices, the FastIron 4802, and the TurboIron/8 use the weighted fair queuing method of packet prioritization by default. To change the method to strict queuing or back to weighted fair queuing, use one of the following methods.

USING THE CLI

To change the queuing method from weighted fair queuing to strict queuing, enter the following commands:

```
BigIron(config)# qos mechanism strict
BigIron(config)# write memory
```

Syntax: [no] qos mechanism strict | weighted

To change the method back to weighted fair queuing, enter the following commands:

```
BigIron(config)# qos mechanism weighted
BigIron(config)# write memory
```

USING THE WEB MANAGEMENT INTERFACE

1. Log on to the device using a valid user name and password for read-write access. The System configuration dialog is displayed.
2. Click on the Weighted or Strict radio button next to QoS.
3. Click the Apply button to save the change to the device's running-config file.

4. Select the Save link at the bottom of the dialog, then select Yes when prompted to save the configuration change to the startup-config file on the device's flash memory.

Configuring the Queues

Each of the four 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 are qosp3, qosp2, qosp1, and qosp0. You can change one or more of the names if desired. To do so, use one of the following methods.

USING THE CLI

To rename queue qosp3 (the premium queue) to "92-octane", enter the following commands:

```
BigIron(config)# qos name qosp3 92-octane
BigIron(config)# write memory
```

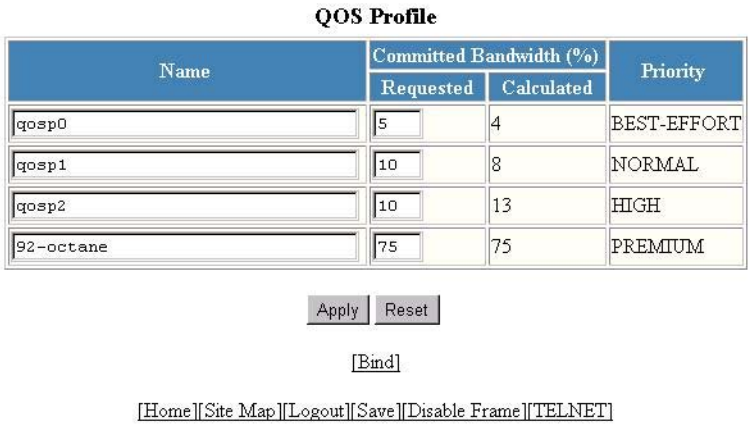
Syntax: qos name <old-name> <new-name>

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.

USING THE WEB MANAGEMENT INTERFACE

1. Log on to the device using a valid user name and password for read-write access. The System configuration dialog is displayed.
2. Click on the plus sign next to Configure in the tree view to expand the list of configuration options.
3. Click on the plus sign next to QoS in the tree view to expand the list of QoS option links.
   - On Layer 2 Switches – Click on the Bind link to display the 802.1q to QOS Profile Binding panel, then click on the Profile link to display the QOS Profile configuration panel.
   - On Layer 3 Switches – Click on the Profile link to display the QoS Profile configuration panel.



4. Edit the strings name the Name fields for the queue(s) you want to rename. In this example, the premium queue is renamed from "qosp3" to "92-octane".
5. Click the Apply button to save the change to the device's running-config file.

6. Select the Save link at the bottom of the dialog, then select Yes when prompted to save the configuration change to the startup-config file on the device's flash memory.

Changing the Minimum Bandwidth Percentages of the Queues

If you are using the weighted fair queuing 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 four QoS queues receive the following minimum guaranteed percentages of a port's total bandwidth.

Queue	Default Minimum Percentage of Bandwidth	
	Chassis devices and the TurboIron/8	**FastIron 4802**
qosp3	80%	75%
qosp2	15%	15%
qosp1	3.3%	5%
qosp0	1.7%	5%

NOTE: The percentages are guaranteed minimum bandwidth percentages. Thus, they apply when a port is fully utilized. When a port is not fully utilized, it is possible for queues to receive more than the configured percentage of bandwidth. You cannot specify a maximum bandwidth percentage for a queue. Any queue can get more than its committed share when other queues are idle.

When the queuing method is weighted fair queuing, 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 fair queuing algorithm.

For example, the default percentages for a Chassis device translate into the following weights.

Queue	Default Minimum Percentage of Bandwidth	Queue Weight
qosp3	80%	4
qosp2	15%	3
qosp1	3.3%	2
qosp0	1.7%	1

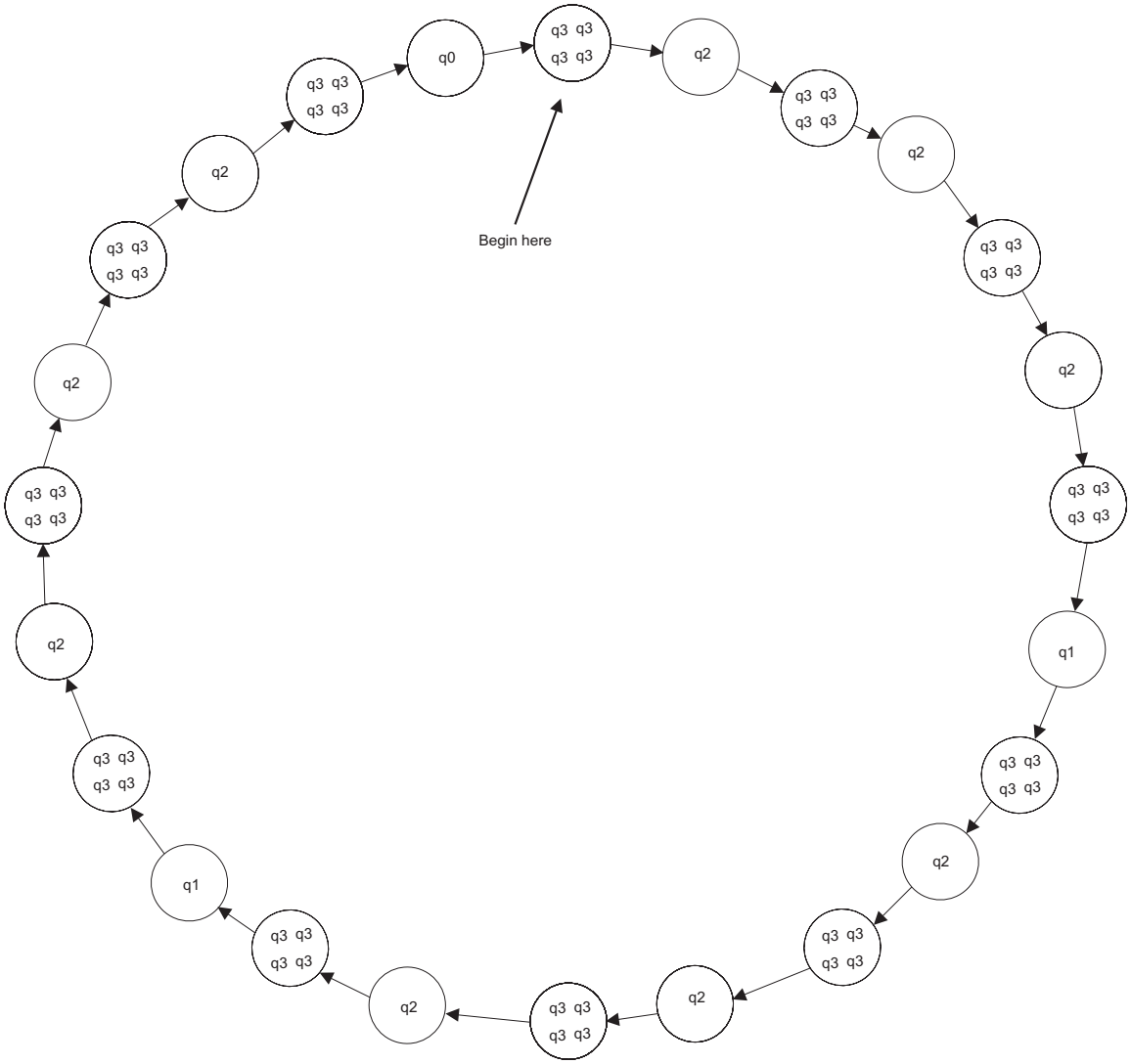
A queue's weight specifies how many packets are sent from the queue each time the queue is serviced. Thus, when the default bandwidth percentages are used, four packets are sent from queue qosp3 each time the queue is serviced, while three packets are sent from queue qosp2 each time it is serviced, and so on. The queuing mechanism interleaves the queues during the cycle so that queue qosp3 is serviced after each visit to any other queue. For example, using the default percentages (and thus the default weights), queue qosp3 receives 12 visits for every one visit to queue qosp0.

The following table shows one full queue cycle using the default bandwidth percentages on a Chassis device.

qosp3 bandwidth % = 80 weight = 4		**qosp2 bandwidth % = 15 weight = 3**		**qosp1 bandwidth % = 3.3 weight = 2**		**qosp0 bandwidth % = 1.7 weight = 1**	
Total visits	**Total packets**	**Total visits**	**Total packets**	**Total visits**	**Total packets**	**Total visits**	**Total packets**
1	4		1				
2	8		2				
3	12	1	3				
4	16				1		
5	20		4				
6	24		5				
7	28	2	6				
8	32			1	2		
9	36		7				
10	40		8				
11	44	3	9				
12	48					1	1

Figure 2.1 illustrates a cycle through the queues.

Figure 2.1 Example of a QoS cycle using the Chassis device default weights



Queue 3: weight=4, minimum percentage=80%

Queue 2: weight=3, minimum percentage=15%

Queue 1: weight=2, minimum percentage=3.3%

Queue 0: weight=1, minimum percentage=1.7%

If you change the percentages for the queues, the software changes the weights, which changes the number of visits a queue receives during a full queue cycle and also the number of packets sent from each queue during each visit. For example, if you change the percentages so that queue qosp3 receives a weight of 5, then the system processes five packets in that queue during each visit to the queue.

NOTE: The weighted fair queuing method is based on packet-level scheduling. As a result, a queue's bandwidth percentage does not necessarily reflect the exact bandwidth share the queue receives. This is due to the effects of variable size packets.

USING THE CLI

To change the minimum guaranteed bandwidth percentages of the queues, enter commands such as the following. Note that this example uses the default queue names.

```
BigIron(config)# qos profile qosp3 75 qosp2 10 qosp1 10 qosp0 5
Profile qosp3     : PREMIUM     bandwidth requested  75% calculated  75%
Profile qosp2     : HIGH        bandwidth requested  10% calculated  13%
Profile qosp1     : NORMAL      bandwidth requested  10% calculated   8%
Profile qosp0     : BEST-EFFORT bandwidth requested   5% calculated   4%
BigIron(config)# write memory
```

Notice that the CLI displays the percentages you request and the percentages the device can provide based on your request. The values are not always the same, as explained below.

Syntax: [no] qos profile <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.

NOTE: The percentages you enter must equal 100. Also, the percentage for the premium queue (the highest priority queue) must be at least 50.

If you enter percentages that are less than the minimum percentages supported for a queue, the CLI recalculates the percentages to fall within the supported minimums. Here is an example. In this example, the values entered for all but the best-effort queue (the lowest priority queue) are much lower than the minimum values supported for those queues.

```
BigIron(config)# qos qosp3 1 qosp2 1 qosp1 2 qosp0 96
Warning - qosp3 bandwidth should be at least 50%
bandwidth scheduling mechanism: weighted priority
Profile qosp3     : PREMIUM     bandwidth requested   1% calculated  50%
Profile qosp2     : HIGH        bandwidth requested   1% calculated  25%
Profile qosp1     : NORMAL      bandwidth requested   2% calculated  13%
Profile qosp0     : BEST-EFFORT bandwidth requested  96% calculated  12%
```

This example shows the warning message that is displayed if you enter a value that is less than 50% for the premium queue. This example also shows the recalculations performed by the CLI. The CLI must normalize the values because the weighted fair queuing algorithm and queue hardware require specific minimum bandwidth allocations. You cannot configure the hardware to exceed the weighted fair queuing limitations.

The CLI normalizes the percentages you enter by increasing the percentages as needed for queues that have less than the minimum percentage, converting the percentages to weights (which the weighted fair queuing algorithm uses), and applying the following equations to calculate the percentages:

qosp3 = w3 / (w3 + 1)

qosp2 = (1 – qosp3) * w2 / (w2 + 1)

qosp1 = (1 – qosp3 – qosp2) * w1 / (w1 + 1)

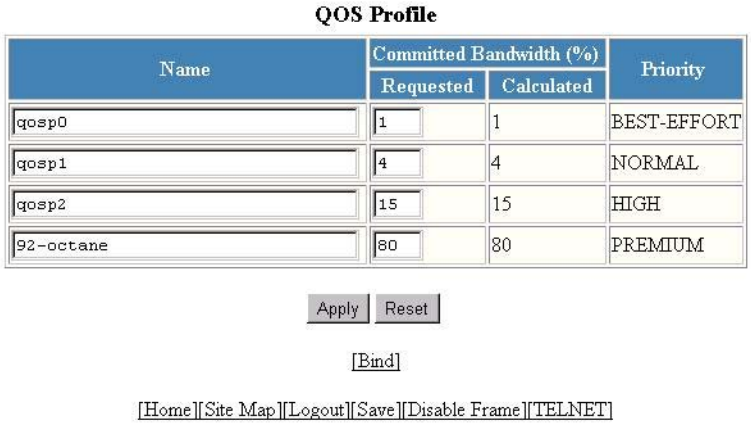
qosp0 = 1 – qosp3 – qosp2 – qosp1

The value “w” stands for “weight”. Thus, these calculations determine the weights that the weighted fair queuing algorithm will use for each queue.

For results that do not differ widely from the percentages you enter, enter successively lower percentages for each queue, beginning with the premium queue. If you enter a higher percentage for a particular queue than you enter for a higher queue, the normalized results can vary widely from the percentages you enter.

USING THE WEB MANAGEMENT INTERFACE

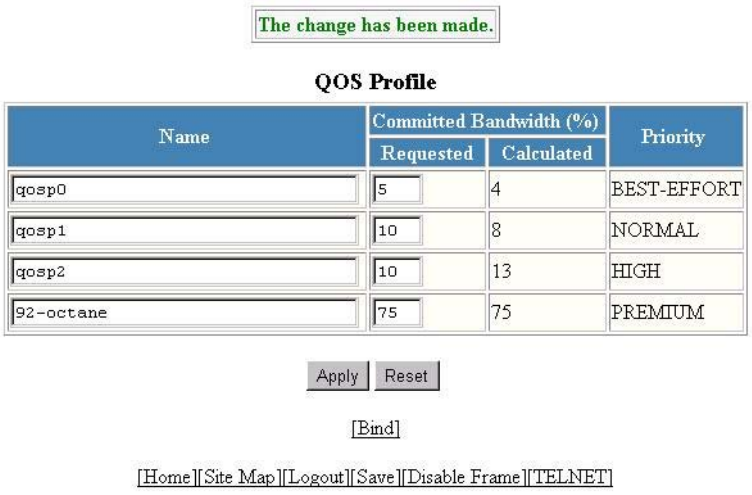
1. Log on to the device using a valid user name and password for read-write access. The System configuration dialog is displayed.
2. Click on the plus sign next to Configure in the tree view to expand the list of configuration options.
3. Click on the plus sign next to QoS in the tree view to expand the list of QoS option links.
   - On Layer 2 Switches – Click on the Bind link to display the 802.1q to QOS Profile Binding panel, then click on the Profile link to display the QOS Profile configuration panel.
   - On Layer 3 Switches – Click on the Profile link to display the QoS Profile configuration panel.



4. Edit the values in the Requested fields for the queue(s) you want to change. In this example, the following minimum bandwidths are requested:
   - qosp0 – 5%
   - qosp1 – 10%
   - qosp2 – 10%
   - 92-octane – 75%

NOTE: The percentages you enter must equal 100. Also, the percentage for the premium queue (the highest priority queue) must be at least 50.

5. Click the Apply button to save the changes to the device's running-config file. Notice that the device calculates the minimum bandwidth percentages that can be allocated to each of the queues based on your percentage requests, and displays the actual percentages in the Calculated column. Here is an example.



6. Select the Save link at the bottom of the dialog, then select Yes when prompted to save the configuration change to the startup-config file on the device's flash memory.

Resetting the Minimum Bandwidth Percentages to Their Defaults

You can use either of the following CLI commands to reset the QoS queues to their default bandwidth percentages (and therefore to their default weights).

USING THE CLI

Enter either of the following commands at the global CONFIG level:

- **qos mechanism weighted**
- **no qos profile**

USING THE WEB MANAGEMENT INTERFACE

You cannot reset the queue profiles to the default bandwidth percentages using the Web management interface.

Displaying the IronClad QoS Profile Configuration

To display the QoS settings, use either of the following methods.

USING THE CLI

To display the QoS settings for all the queues, enter the following command from any level of the CLI:

```
BigIron(config)# show qos-profiles all
bandwidth scheduling mechanism: weighted priority
Profile qosp3     : PREMIUM     bandwidth requested  75% calculated  75%
Profile qosp2     : HIGH        bandwidth requested  10% calculated  13%
Profile qosp1     : NORMAL      bandwidth requested  10% calculated   8%
Profile qosp0     : BEST-EFFORT bandwidth requested   5% calculated   4%
```

Syntax: show qos-profiles all | <name>

The **all** parameter displays the settings for all four queues. The <name> parameter displays the settings for the specified queue.

USING THE WEB MANAGEMENT INTERFACE

1. Log on to the device using a valid user name and password for read-write access. The System configuration dialog is displayed.
2. Click on the plus sign next to Configure in the tree view to expand the list of configuration options.
3. Click on the plus sign next to QoS in the tree view to expand the list of QoS option links.
4. Click on the Bind link to display the 802.1q to QOS Profile Binding panel.

Assigning QoS Priorities to Traffic

By default, traffic in the following categories is forwarded using the best-effort queue (qosp0) on Chassis devices, the FastIron 4802, or the TurboIron/8. Traffic in these categories is forwarded by default using the normal queue on Stackable devices:

- Incoming port (sometimes called the ingress port)
- Port-based VLAN membership
- Static destination MAC entry
- Layer 3 and Layer 4 information (IP and TCP/UDP source and destination information)
- AppleTalk socket

The following sections describe how to change the priority for each of the items listed above.

NOTE: Tagged VLAN traffic is placed in a queue corresponding to the 802.1p priority in the tag by default. Thus, if a tagged packet contains priority 7 in the tag (corresponding to the premium queue), the device places this packet in the premium queue of the packet's outbound port. On Chassis devices, the FastIron 4802, and the TurboIron/8, you can change or remove the effect of the 802.1p priority in the tags by reassigning the priority levels to different queues. See "Reassigning 802.1p Priorities to Different Queues" on page 2-14.

Although it is possible for a packet to qualify for an adjusted QoS priority based on more than one of the criteria above, the system always gives a packet the highest priority for which it qualifies. Thus, if a packet on a Chassis device 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 either specify a number from 0 – 7 (Chassis devices, the FastIron 4802, and the TurboIron/8) or specify "high" or normal (Stackable devices). On Chassis devices, the FastIron 4802, and the TurboIron/8, the priority number specifies the IEEE 802.1 equivalent to one of the four Foundry QoS queues. The numbers correspond to the queues as follows.

Priority Level	Queue
6, 7	qosp3
4, 5	qosp2
2, 3	qosp1
0, 1	qosp0

On Stackable devices, the device processes all packets in a port's high priority queue before processing any packets in the port's normal queue.

Changing a Port's Priority

To change a port's QoS priority, use one of the following methods. The priority applies to inbound traffic on the port.

USING THE CLI

To change the QoS priority of port 1/1 on a Chassis device to the high queue (qosp2), enter the following commands:

```
BigIron(config)# interface ethernet 1/1
BigIron(config-if-1/1)# priority 5
BigIron(config-if-1/1)# write memory
```

Syntax: [no] priority <num>

The <num> parameter can be from 0 – 7 and specifies the IEEE 802.1 equivalent to one of the four QoS queues.

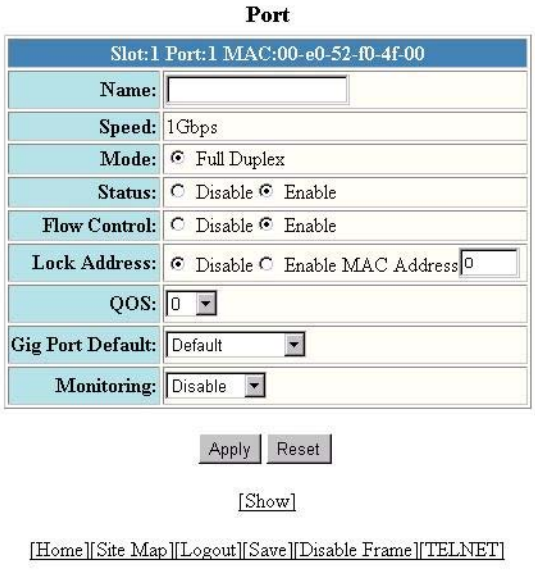
To change the QoS priority of port 1 on a Stackable device to the high queue, enter the following commands:

```
NetIron(config)# interface ethernet 1
NetIron(config-if-1)# priority high
NetIron(config-if-1)# write memory
```

Syntax: [no] priority high | normal

USING THE WEB MANAGEMENT INTERFACE

1. Log on to the device using a valid user name and password for read-write access. The System configuration dialog is displayed.
2. Click on the plus sign next to Port in the tree view to display the configuration options.
3. Select the link to the port type you want (for example, Ethernet) to display the Port table.
4. Scroll down to the port for which you want to change the QoS level, then click on the Modify button to the right of the port information to display the Port configuration panel, as shown in the following example.



5. Select the QoS level:
   - On a Chassis device, the FastIron 4802, or the TurboIron/8, select a number from 0 – 7 from the QoS field's pulldown menu.
   - On a Stackable device, select high or normal from the QoS field's pulldown menu.

6. Click the Apply button to save the change to the device's running-config file.
7. Select the Save link at the bottom of the dialog, then select Yes when prompted to save the configuration change to the startup-config file on the device's flash memory.

Changing a Layer 2 Port-Based VLAN's Priority

By default, VLANs have priority 0 (Chassis devices, the FastIron 4802, and the TurboIron/8) or normal (Stackable devices). To change a port-based VLAN's QoS priority, use one of the following methods. The priority applies to outbound traffic on ports in the VLAN.

NOTE: Tagged packets also contain a priority value in the 802.1q tag. If you use the default priority for a VLAN, a tagged packet that exits on that VLAN can be placed into a higher priority queue based on the port priority, the priority in the 802.1q tag, and so on. If you do not want the device to make priority decisions based on 802.1q tags, you can change the priority for 802.1q tags on a VLAN basis on Chassis devices, the FastIron 4802, or the TurboIron/8. See "Reassigning 802.1p Priorities to Different Queues" on page 2-14".

USING THE CLI

To change the QoS priority of port-based VLAN 20 on a Chassis device to the premium queue (qosp3), enter the following commands:

```
BigIron(config)# vlan 20
BigIron(config-vlan-20)# priority 7
BigIron(config-vlan-20)# write memory
```

Syntax: [no] priority <num>

The <num> parameter can be from 0 – 7 and specifies the IEEE 802.1 equivalent to one of the four QoS queues.

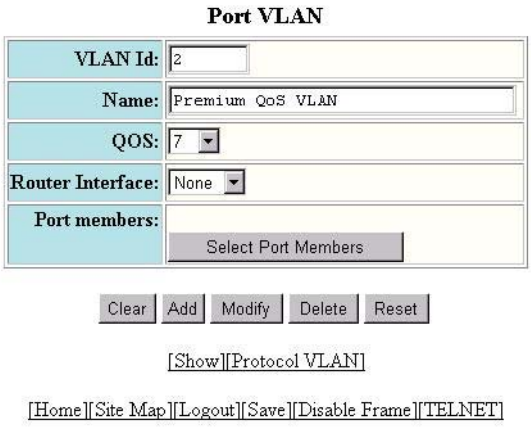
To change the QoS priority of port-based VLAN 20 on a Stackable device to the high queue, enter the following commands:

```
NetIron(config)# vlan 20
NetIron(config-vlan-20)# priority high
NetIron(config-vlan-20)# write memory
```

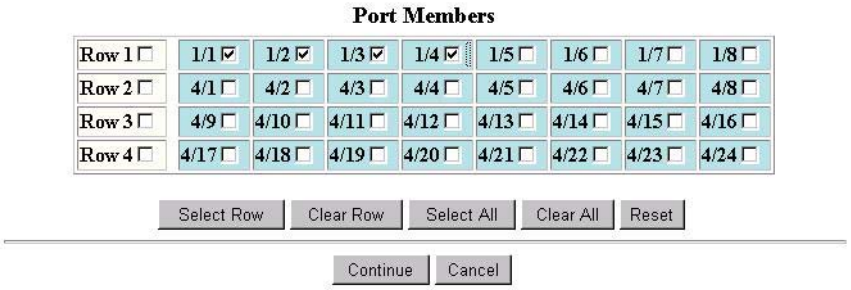
Syntax: [no] priority high | normal

USING THE WEB MANAGEMENT INTERFACE

1. Log on to the device using a valid user name and password for read-write access. The System configuration dialog is displayed.
2. Click on the plus sign next to Configure in the tree view to expand the list of configuration options.
3. Click on the plus sign next to VLAN in the tree view to expand the list of VLAN option links.
4. Click on the Port link to display the Port VLAN panel.
   - If you are adding a new port-based VLAN, click on the Add Port VLAN link to display the Port VLAN configuration panel, as shown in the following example.
   - If you are modifying an existing port-based VLAN, click on the Modify button to the right of the row describing the VLAN to display the Port VLAN configuration panel, as shown in the following example.



5. Select the QoS level:
   - On a Chassis device, the FastIron 4802, or the TurboIron/8, select a number from 0 – 7 from the QoS field's pulldown menu.
   - On a Stackable device, select high or normal from the QoS field's pulldown menu.
6. If you are adding a new VLAN, click the Select Port Members button to display the Port Members dialog, as shown in the following example.



7. Select the ports you are placing in the VLAN. To select a row, click on the checkbox next to the row number, then click on the Select Row button.
8. When you finish selecting the ports, click on the Continue button to return to the Port VLAN configuration dialog.
9. Click the Add button (to add a new VLAN) or the Modify button (if you are modifying an existing VLAN) to save the change to the device's running-config file.
10. Select the Save link at the bottom of the dialog, then select Yes when prompted to save the configuration change to the startup-config file on the device's flash memory.

Reassigning 802.1p Priorities to Different Queues

Tagged priority applies to tagged packets that come in from tagged ports. These packets have a tag in the header that specifies the packet's VLAN ID and its 802.1p priority tag value, which is 3 bits long.

NOTE: This section applies to Chassis devices, the FastIron 4802, and the TurboIron/8 only.

By default, a Foundry device interprets the prioritization information in the 3-bit priority tag as follows.

Priority Level	Queue
6, 7	qosp3
4, 5	qosp2
2, 3	qosp1
0, 1	qosp0

This is the Foundry default interpretation for the eight prioritization values in every context (VLAN, static MAC entry, IP access policy, and so on). If the VLAN for the packet uses the default priority (0, equal to the qosp0 queue), then the Foundry device uses the priority information in the packet to assign the packet to a queue on its incoming port. However, if the VLAN or the incoming port itself has a higher priority than the packet's 802.1p priority, the Foundry device uses the VLAN priority or incoming port priority, whichever is higher.

You can specify how the Foundry device interprets the 3-bit priority information by reassigning the priority levels to other queues. For example, if you want the device to disregard the 802.1p priority and instead assign the priority based on other items (VLAN, port, and so on), configure the device to set all the 802.1p priorities to the best-effort queue (qosp0). If a tagged packet's 802.1p priority level is always in the qosp0 queue, then the packet's outbound queue is affected by other items such as incoming port, VLAN, and so on.

To reassign the priorities to different queues, use either of the following methods.

USING THE CLI

To reassign all 802.1p priority levels 2 – 7 to the best-effort queue (qosp0), enter the following commands:

```
BigIron(config)# qos tagged-priority 2 qosp0
BigIron(config)# qos tagged-priority 3 qosp0
BigIron(config)# qos tagged-priority 4 qosp0
BigIron(config)# qos tagged-priority 5 qosp0
BigIron(config)# qos tagged-priority 6 qosp0
BigIron(config)# qos tagged-priority 7 qosp0
BigIron(config)# write memory
```

Syntax: [no] qos tagged-priority <num> <queue>

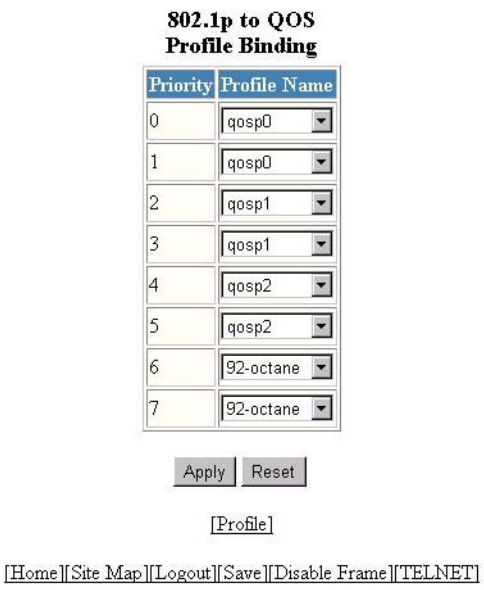
The <num> parameter can be from 0 – 7 and specifies the IEEE 802.1 equivalent to one of the four QoS queues.

The <queue> parameter specifies the queue to which you are reassigning the priority level. You must specify one of the named queues. The default names are qosp3, qosp2, qosp1, and qosp0. The example above reassigns the 802.1p levels to queue qosp0. (There is no need to reassign levels 0 and 1 in this case, because they are already assigned to qosp0 by default.)

USING THE WEB MANAGEMENT INTERFACE

1. Log on to the device using a valid user name and password for read-write access. The System configuration dialog is displayed.
2. Click on the plus sign next to Configure in the tree view to expand the list of configuration options.
3. Click on the plus sign next to QoS in the tree view to expand the list of QoS option links.

4. Click on the Bind link to display the QoS 802.1p to QoS Profile Binding configuration panel, as shown in the following figure.



5. For each priority level, select the QoS queue to which you want to reassign the profile by selecting the queue name from the Profile field's pulldown list. For example, to reassign priority 7 to QoS queue qosp0, select qosp0 from the Profile Name field's pulldown list in the row for priority 7.
6. Click the Apply button to save the change to the device's running-config file.
7. Select the Save link at the bottom of the dialog, then select Yes when prompted to save the configuration change to the startup-config file on the device's flash memory.

Displaying the Queue Assignments for the 802.1p Priorities

To display the queues to which the 802.1p priorities are assigned, use either of the following methods.

USING THE CLI

To display the queue assignments for all the priorities, enter the following command at any level of the CLI:

```
BigIron(config)# show priority-mapping all
802.1p priority 0 mapped to qos profile qosp0
802.1p priority 1 mapped to qos profile qosp0
802.1p priority 2 mapped to qos profile qosp1
802.1p priority 3 mapped to qos profile qosp1
802.1p priority 4 mapped to qos profile qosp2
802.1p priority 5 mapped to qos profile qosp2
802.1p priority 6 mapped to qos profile qosp3
802.1p priority 7 mapped to qos profile qosp3
```

In this example, the priorities still have their default queue assignments.

Syntax: show priority-mapping all | <num>

The **all** parameter displays the queue assignments for all the priorities. Alternatively, you can display the assignment for a particular level by specifying the level number, as shown in the following example.

```
BigIron(config)# show priority-mapping 1
802.1p priority 1 mapped to qos profile qosp0
```


USING THE WEB MANAGEMENT INTERFACE

1. Log on to the device using a valid user name and password for read-write access. The System configuration dialog is displayed.
2. Click on the plus sign next to Configure in the tree view to expand the list of configuration options.
3. Click on the plus sign next to QoS in the tree view to expand the list of QoS option links.
4. Click on the Bind link to display the QoS 802.1p to QoS Profile Binding configuration panel. The queue assignments are listed for each of the eight priority levels.

Assigning Static MAC Entries to Priority Queues

By default, all MAC entries are in the best effort queue (Chassis devices, the FastIron 4802, and the TurboIron/8) or the normal queue (Stackable devices). When you configure a static MAC entry, you can assign the entry to a higher QoS level using either of the following methods.

USING THE CLI

To configure a static MAC entry and assign the entry to the premium queue on a Chassis device, enter commands such as the following:

```
BigIron(config)# vlan 9
BigIron(config-vlan-9)# static-mac-address 1145.1163.67FF ethernet 1/1 priority 7
BigIron(config-vlan-9)# write memory
```

Syntax: [no] static-mac-address <mac-addr> ethernet <portnum> [priority <num>]
[host-type | router-type | fixed-host]

The <num> parameter can be from 0 – 7 and specifies the IEEE 802.1 equivalent to one of the four 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 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.

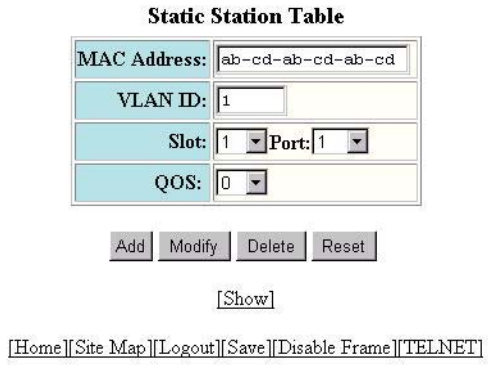
To configure a static MAC entry and assign the entry to the high queue on a Stackable device, enter commands such as the following:

```
FastIron(config)# vlan 9
FastIron(config-vlan-9)# static-mac-address 1145.1163.67FF ethernet 1 high-priority
FastIron(config-vlan-9)# write memory
```

Syntax: static-mac-address <mac-addr> ethernet <portnum> [normal-priority | high-priority]
[host-type | router-type]

USING THE WEB MANAGEMENT INTERFACE

1. Log on to the device using a valid user name and password for read-write access. The System configuration dialog is displayed.
2. Click on the plus sign next to Configure in the tree view to expand the list of configuration options.
3. Select the Static Station link to display the Static Station Table.
   - If the system already contains static MAC addresses and you are adding a new static MAC address, click on the Add Static Station link to display the Static Station Table configuration panel, as shown in the following example.
   - If you are modifying an existing static MAC address, click on the Modify button to the right of the row describing the static MAC address to display the Static Station Table configuration panel, as shown in the following example.



4. Enter or edit the MAC address, if needed. Specify the address in the following format: xx-xx-xx-xx-xx-xx.
5. Change the VLAN number if needed by editing the value in the VLAN ID field.
6. Select the port number from the Slot (for Chassis devices) and Port pulldown lists.
7. Select the QoS level:
   - On a Chassis device, the FastIron 4802, or the TurboIron/8, select a number from 0 – 7 from the QoS field's pulldown menu.
   - On a Stackable device, select high or normal from the QoS field's pulldown menu.
8. Click the Add button (to add a new static MAC entry) or the Modify button (if you are modifying an existing entry) to save the change to the device's running-config file.
9. Click the Apply button to save the change to the device's running-config file.
10. Select the Save link at the bottom of the dialog, then select Yes when prompted to save the configuration change to the startup-config file on the device's flash memory.

Assigning IP and Layer 4 Sessions to Priority Queues

You can assign specific traffic flows to queues by configuring IP access policies. Beginning with software release 07.6.01, you can also use ACLs to prioritize traffic flows on all Layer 3 Switches, except on the NetIron 4802, FastIron I, and FastIron II. Foundry recommends that you use the priority options in extended ACLs to prioritize traffic on supported Layer 3 Switches. (See "QoS Options for IP ACLs (Rule-Based ACLs)" on page 6-44 for details.)

This section presents information on the IP access policies. IP access policies allow you to assign flows to priority queues based on any combination of the following criteria:

- Source IP address
- Destination IP address
- Layer 4 type (TCP or UDP)
- TCP or UDP port number

You configure IP access policies globally, then apply them to specific ports. QoS policies apply only to outbound traffic, so you must apply the QoS polices to a port's outbound direction instead of the port's inbound direction.

To configure an IP access policy for assigning a traffic flow to a priority queue, use either of the following methods.

USING THE CLI

The CLI syntax differs between Layer 3 Switches and Layer 2 Switches. Examples and syntax are shown for both types of devices.

Layer 3 Switch Syntax

To assign a priority of 4 to all HTTP traffic on port 3/12 on a BigIron Layer 3 Switch, enter the following:

```
BigIron(config)# ip access-policy 1 priority 4 any any tcp eq http
BigIron(config)# int e 3/12
BigIron(config-if-3/12)# ip access-policy-group out 1
```

Here is the syntax for chassis Layer 3 Switches.

Syntax: [no] ip access-policy <num> priority <0-7> <ip-addr> <ip-mask> | any <ip-addr> <ip-mask> | any icmp | igmp | igrp | ospf | tcp | udp | <num> [<operator> [<tcp/udp-port-num>]]

Syntax: ip access-policy-group in | out <policy-list>

Here is the syntax for stackable Layer 3 Switches.

Syntax: ip access-policy <num> high | normal <ip-addr> <ip-mask> | any <ip-addr> <ip-mask> | any tcp | udp [<operator> [<tcp/udp-port-num>]]

Syntax: ip access-policy-group in | out <policy-list>

The <num> parameter is the policy number.

The **priority** <0-7> parameter on Chassis devices specifies the QoS priority level. The default is 0 (best effort, qosp0). The highest priority is 7 (premium, qosp3).

The **high | normal** parameter on Stackable devices specifies the QoS priority level. The default is **normal**.

The <ip-addr> <ip-mask> **| any** <ip-addr> <ip-mask> **| any** parameters specify the source and destination IP addresses. If you specify a particular IP address, you also need to specify the mask for that address. If you specify **any** to apply the policy to all source or destination addresses, you do not need to specify **any** again for the mask. Make sure you specify a separate address and mask or any for the source and destination address.

The **icmp | igmp | igrp | ospf | tcp | udp |** <num> parameter specifies the Layer 4 port to which you are applying the policy. If you specify **tcp** or **udp**, you also can use the optional <operator> and <tcp/udp-port-num> parameters to fine-tune the policy to apply to specific TCP or UDP ports.

The <operator> parameter applies only if you use the **tcp** or **udp** parameter above. Use the <operator> parameter to specify the comparison condition for the specific TCP or UDP ports. For example, if you are configuring QoS 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**.
- **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.
- **established** – This operator applies only to TCP packets. If you use this operator, the QoS 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.

The **in** parameter applies the policy to packets received in the port.

The **out** parameter applies the policy to packets sent on the port.

NOTE: To apply the policy to traffic in both directions, enter two **ip access-policy-group** commands, one specifying the **in** parameter, and the other specifying the **out** parameter.

The <policy-list> parameter is a list of policy IDs.

NOTE: The device applies the policies in the order you list them, so make sure you order them in such a way that you receive the results you expect. Once a packet matches a policy, the device takes the action specified in that policy and stops comparing the packet to the policies in the list.

Figure 2.2 and Figure 2.3 show the CLI syntax for configuring a Layer 4 QoS policy on a Foundry Layer 3 Switch. Notice that the syntax differs slightly depending on whether you are configuring a Stackable Layer 3 Switch, or a Chassis Layer 3 Switch, a FastIron 4802 Layer 3 Switch, or a TurboIron/8 Layer 3 Switch.

Figure 2.2 QoS IP policy syntax for a Foundry router (1 of 2)

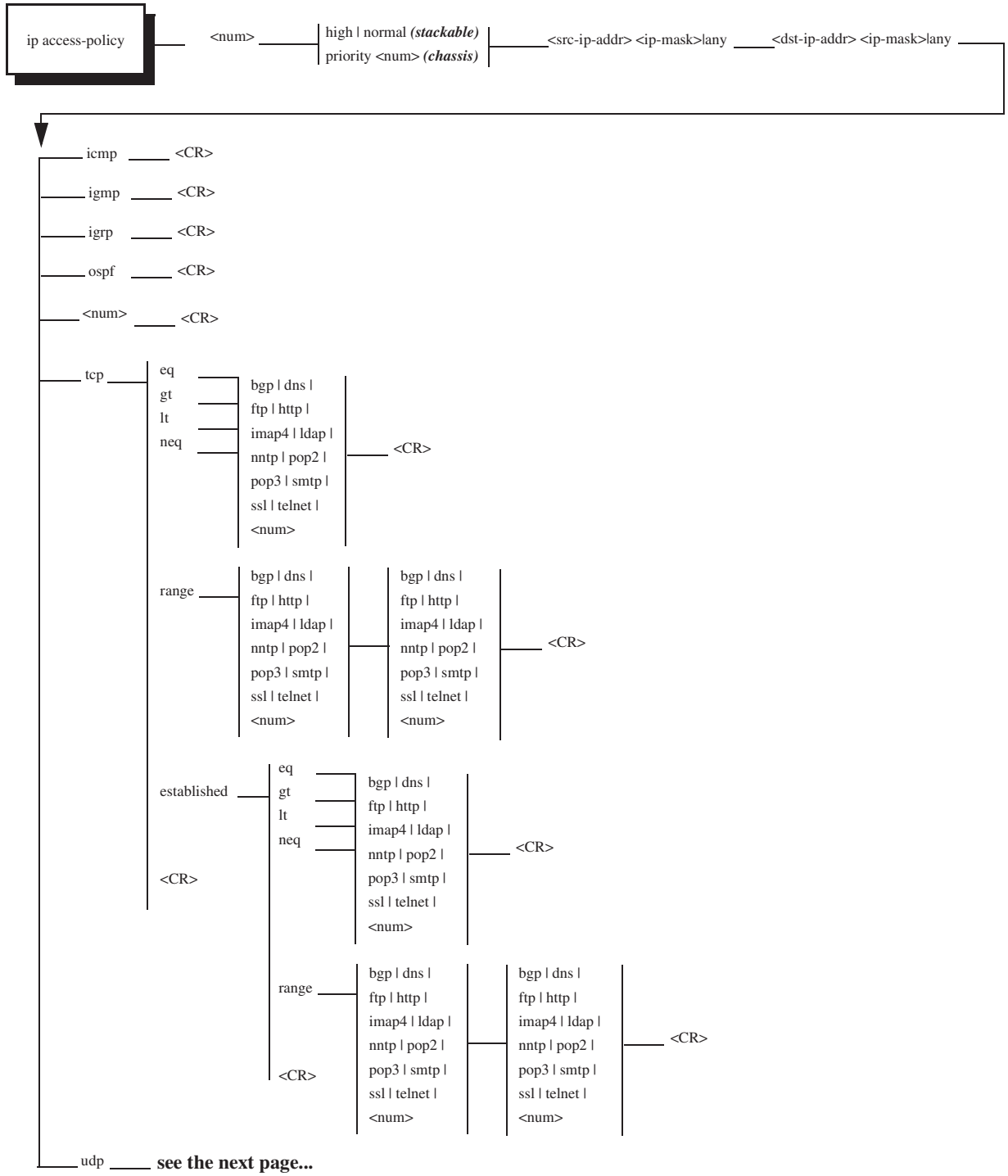
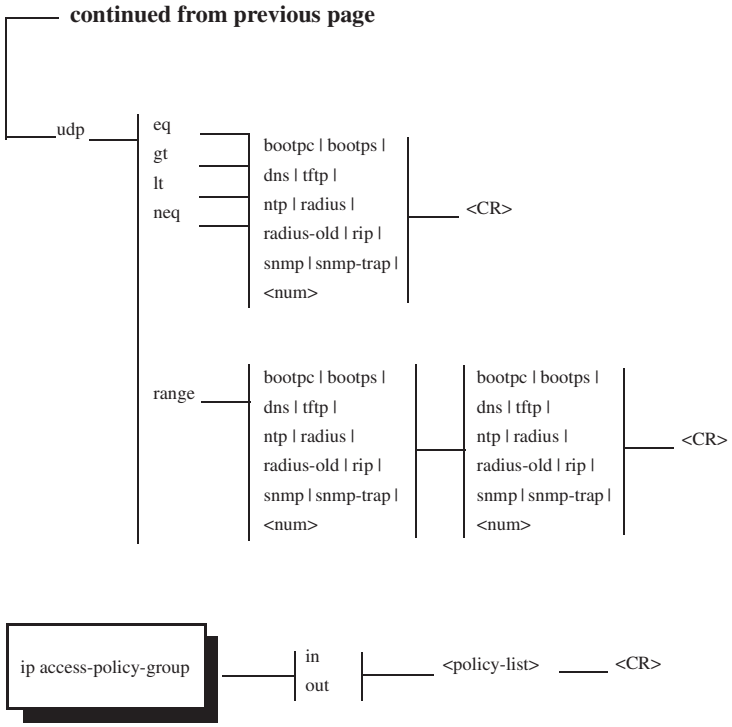


Figure 2.3 QoS IP policy syntax for a Foundry router (2 of 2)



Layer 2 Switch Syntax

To assign a priority of 7 to FTP traffic on all ports on a FastIron II Layer 2 Switch, enter the following commands:

```
FastIron(config)# ip policy 1 7 tcp ftp global
FastIron(config)# write memory
```

To assign a priority of 7 to HTTP traffic on ports 1/1 and 1/2 only, enter the following commands:

```
FastIron(config)# ip policy 2 7 tcp http local
FastIron(config)# int ethernet 1/1
FastIron(config-if-1/1)# ip-policy 2
FastIron(config-if-1/1)# int ethernet 1/2
FastIron(config-if-1/2)# ip-policy 2
FastIron(config)# write memory
```

Here is the syntax for Chassis Layer 2 Switches.

Syntax: [no] ip policy <num> priority <0-7> tcp | udp <tcp/udp-port-num> global | local

Syntax: [no] ip policy <num> high | normal tcp | udp <tcp/udp-port-num> global | local

Syntax: [no] ip-policy <num>

The <num> parameter is the policy number.

The **priority** <0-7> parameter on Chassis devices specifies the QoS priority level. The default is 0 (best effort, qosp0). The highest priority is 7 (premium, qosp3).

The **high | normal** parameter on Stackable devices specifies the QoS priority level. The default is **normal**.

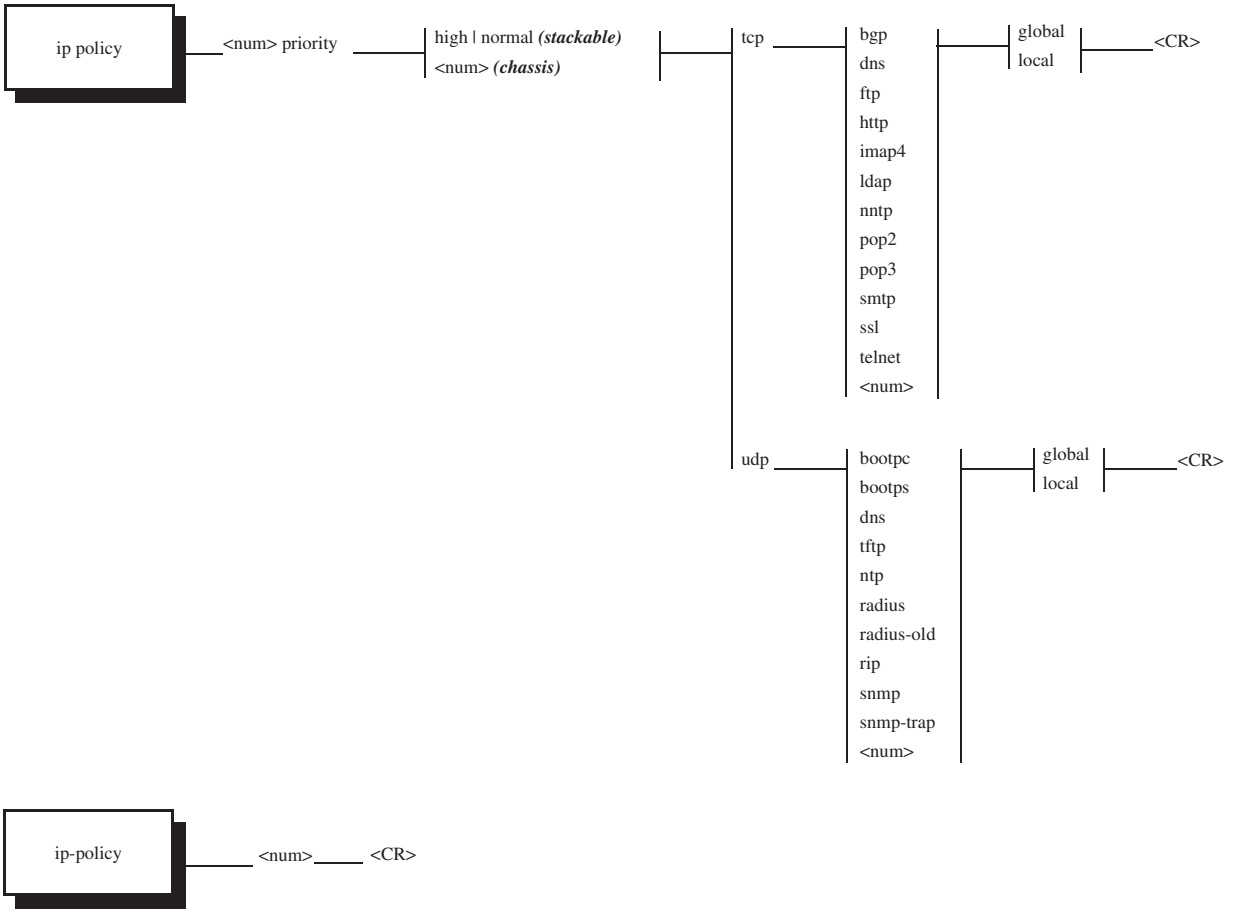
The **tcp | udp** <tcp/udp-port-num> parameter specifies the TCP or UDP port to which you are applying the policy.

The **global** and **local** parameters specify the scope of the policy:

- If you specify **global**, the policy applies to all ports.
- If you specify **local**, the policy will apply to the ports you specify. Use the following command on the Interface level of the CLI to apply the policy to a port: **ip-policy** <num>

Figure 2.4 shows the CLI syntax for configuring a QoS policy on a Foundry Layer 2 Switch. The value "<CR>" means "carriage return", also known as the Enter key.

Figure 2.4 QoS IP policy syntax for a Foundry Layer 2 Switch



NOTE: The **ip policy** command allows you to configure global or local QoS policies. Use the **ip-policy** command (note the difference between "**ip policy**" and "**ip-policy**") at the Interface level of the CLI to apply a local policy to a specific interface.

USING THE WEB MANAGEMENT INTERFACE

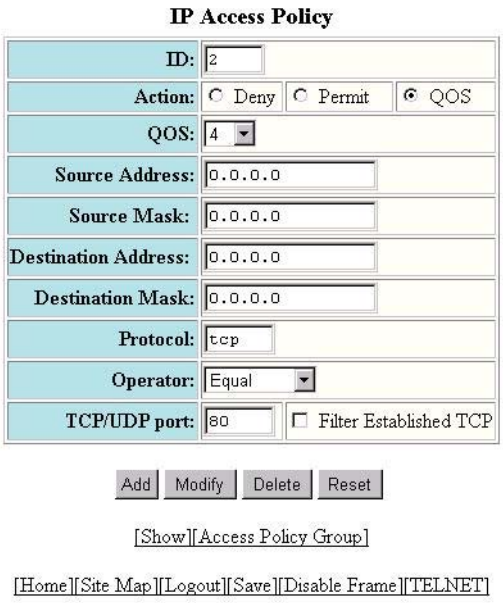
The Web management options for assigning QoS priorities to traffic flows differ between Layer 3 Switches and Layer 2 Switches. Examples are shown for both types of devices.

Layer 3 Switch

To assign a priority of 4 to all HTTP traffic on port 3/12 on a BigIron Layer 3 Switch, perform the following steps:

1. Log on to the device using a valid user name and password for read-write access. The System configuration dialog is displayed.

2. Click on the plus sign next to Configure in the tree view to expand the list of configuration options.
3. Click on the plus sign next to IP in the tree view to expand the list of IP option links.
4. Click on the Access Policy link to display the IP Access Policy panel.
   - If the system already contains IP access policies and you are adding a new one, click on the Add IP Access Policy link to display the IP Access Policy configuration panel, as shown in the following example.
   - If you are modifying an existing IP access policy, click on the Modify button to the right of the row describing the IP access policy to display the IP Access Policy configuration panel, as shown in the following example.



5. Enter the ID for the policy in the ID field.
6. Select the QoS radio button next to Action.
7. Select the QoS level:
   - On a Chassis device, the FastIron 4802, or the TurboIron/8, select a number from 0 – 7 from the QoS field's pulldown menu.
   - On a Stackable device, select high or normal from the QoS field's pulldown menu.
8. Enter the source IP address and network mask in the Source Address and Source Mask fields. To specify "any" for a field, leave all four zeros in the field. In this example, leave the zeros.
9. Enter the destination IP address and network mask in the Destination Address and Destination Mask fields. To specify "any" for a field, leave all four zeroes in the field. In this example, leave the zeros.
10. If you want the policy to apply only to packets containing specific types of Layer 4 traffic, enter the protocol in the Protocol field. You can enter the protocol's Layer 4 port number or one of the following well-known names:
    - icmp
    - igmp
    - igrp
    - ospf
    - tcp

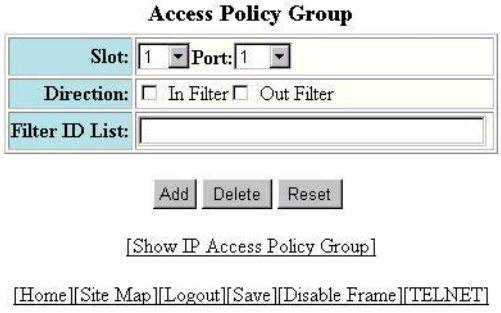
- udp

In this example, enter tcp.

11. If you entered tcp or udp, you also can select one of the following comparison operators from the Operator field.
    - Equal – The policy applies to the TCP or UDP port name or number you enter in the TCP/UDP port field. In this example, select Equal.
    - Greater – The policy applies to TCP or UDP port numbers greater than the port number or the numeric equivalent of the port name you enter in the TCP/UDP port field.
    - Less – 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 in the TCP/UDP port field.
    - Not Equal – The policy applies to all TCP or UDP port numbers except the port number or port name you enter in the TCP/UDP port field.
12. If you entered tcp or udp in the Protocol field, enter the TCP or UDP port number in the TCP/UDP port field. In this example, enter 80 (the well-known port for HTTP).
13. If you entered tcp in the Protocol field and you want the policy to apply to TCP sessions that are already in effect, click on the checkbox next to Established. If you select this option, the QoS 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 option applies only to destination TCP ports, not to source TCP ports.

14. Click the Add button (to add a new policy) or the Modify button (if you are modifying an existing policy) to save the policy to the device's running-config file.
15. Select the Access Policy Group link to display the Access Policy Group panel.
    - If the system already contains IP access policy groups and you are adding a new one, click on the Add IP Access Policy Group link to display the IP Access Policy Group configuration panel, as shown in the following example.
    - If you are modifying an existing IP access policy, click on the Modify button to the right of the row describing the IP access policy group to display the IP Access Policy Group configuration panel, as shown in the following example.



16. Select the port number from the Slot (for Chassis devices) and Port pulldown lists. In this example, select 3/12.
17. Click the checkbox next to In Filter, Out Filter, or next to both options to indicate the traffic direction to which you are applying the policy.
    - The In Filter option applies the policy to packets received in the port.

- The Out Filter option applies the policy to packets sent on the port.
- If you select both, the policy applies to traffic in both directions.

In this example, select Out Filter.

18. Enter the policy IDs in the Filter ID List field.

NOTE: The device applies the policies in the order you list them, so make sure you order them in such a way that you receive the results you expect. Once a packet matches a policy, the device takes the action specified in that policy and stops comparing the packet to the policies in the list.

19. Click the Add button to apply the change to the device's running-config file.
20. Select the Save link at the bottom of the dialog, then select Yes when prompted to save the configuration change to the startup-config file on the device's flash memory.

NOTE: You also can access the dialog for saving configuration changes by clicking on Command in the tree view, then clicking on Save to Flash.

Layer 2 Switch

To assign a priority of 7 to FTP traffic on all ports on a FastIron II Layer 2 Switch, perform the following steps:

1. Log on to the device using a valid user name and password for read-write access. The System configuration dialog is displayed.
2. Select the Layer 4 QoS link to display the QoS panel.
3. Enter the ID for the policy in the ID field.
4. Select the Switch or Port radio button next to Scope to indicate whether the policy applies globally or only to certain ports.
5. Select the QoS level:
   - On a Chassis device, the FastIron 4802, or the TurboIron/8, select a number from 0 – 7 from the QoS field's pulldown menu. In this example, select 7.
   - On a Stackable device, select high or normal from the QoS field's pulldown menu.
6. Select the UDP or TCP radio button next to Protocol to specify the type of traffic to which the QoS policy applies.
7. Select a well-known TCP or UDP port name (depending on whether you selected TCP or UDP) from the TCP/UDP Port field's pulldown list. To enter a port number instead, click on the User Define button to change the field into an entry field, then enter the port number. For this example, select FTP.
8. Click the Add button to apply the change to the device's running-config file.
9. If you selected Port in step 4, click on Port QoS to display the Port QoS panel. Otherwise, go to step 13.
10. Select the port number from the Slot (for Chassis devices) and Port pulldown lists.
11. Enter the policy IDs in the QoS ID List field.

NOTE: The device applies the policies in the order you list them, so make sure you order them in such a away that you receive the results you expect. Once a packet matches a policy, the device takes the action specified in that policy and stops comparing the packet to the policies in the list.

12. Click the Add button to apply the change to the device's running-config file.
13. Select the Save link at the bottom of the dialog, then select Yes when prompted to save the configuration change to the startup-config file on the device's flash memory.

Assigning AppleTalk Sockets to Priority Queues

By default, all AppleTalk sockets are in the best effort queue (Chassis devices, the FastIron 4802, or the TurboIron/8) or the normal queue (Stackable devices). To assign an AppleTalk socket to a higher priority queue, use either of the following methods.

USING THE CLI

To assign socket 123 to the premium queue on a Chassis device, enter the following commands:

```
BigIron(config)# appletalk qos socket 123 priority 7
BigIron(config)# write memory
```

Here is the syntax for Chassis Layer 3 Switches.

Syntax: [no] appletalk qos socket <num> priority <num>

Here is the syntax for Stackable Layer 3 Switches.

Syntax: [no] appletalk qos socket <num> high | normal

The first <num> parameter specifies the socket number.

The second <num> parameter (Chassis devices, the FastIron 4802, or the TurboIron/8) can be from 0 – 7 and specifies the IEEE 802.1 equivalent to one of the four QoS queues.

The **high | normal** parameter (Stackable devices) indicates the priority level.

USING THE WEB MANAGEMENT INTERFACE

1. Log on to the Layer 3 Switch using a valid user name and password for read-write access. The System configuration dialog is displayed.
2. If AppleTalk is not already enabled, enable it by selecting the Enable radio button next to AppleTalk, then clicking Apply.
3. Click on the plus sign next to Configure in the tree view to expand the list of configuration options.
4. Click on the plus sign next to AppleTalk in the tree view to expand the list of AppleTalk option links.
5. Click on the Socket QoS link to display the AppleTalk Socket QoS panel, as shown in the following example.

6. Edit the socket number in the Socket field if needed.
7. Select the QoS level:
   - On a Chassis device, the FastIron 4802, or the TurboIron/8, select a number from 0 – 7 from the QoS field's pulldown menu.
   - On a Stackable device, select high or normal from the QoS field's pulldown menu.
8. Click on the Apply button to apply the new QoS setting to the socket number specified in the Socket field or click on the Apply To All Sockets button to apply the new QoS setting to all AppleTalk sockets.

9. Select the Save link at the bottom of the dialog, then select Yes when prompted to save the configuration change to the startup-config file on the device's flash memory.

IP ToS-Based QoS

NOTE: The ToS-based QoS described in this chapter applies only to the NetIron stackable Layer 3 Switch. To configure ToS-based QoS on a JetCore device, VM1 module, or 10 Gigabit Ethernet module, see "Configuring Enhanced Quality of Service" on page 4-1. To configure ToS-based QoS on a FastIron Edge Switch, see "Configuring Quality of Service on a FastIron Edge Switch and FastIron Edge Switch X-Series" on page 3-1

You can configure a device to use the value in IP packet's Type of Service (ToS) field to prioritize forwarding of the packet through the Foundry device. In addition, you can mark an outbound packet with an 802.1Q priority, a Differentiated Service codepoint (DSCP), or both.

An IP version 4 packet can contain prioritization information that specifies the service the packet should expect when travelling through a network. For example, the packet can contain prioritization information in the following places:

- The IEEE frame can contain an 802.1Q priority (a value from 0 – 7).
- The Type of Service (ToS) field in the IP header can contain values that a forwarding device can interpret as one of the following:
  - IP precedence – The forwarding device can interpret the three most-significant bits (0 – 2) in the ToS field as an IP precedence value. The IP precedence values are defined in RFC 791.
  - Differentiated Service codepoint (DSCP) – The forwarding device can interpret the six most-significant bits (0 – 5) in the ToS field as a DSCP, as defined in RFCs 2474 and 2475.

A device running a software release earlier than 07.1.16 can prioritize a packet based on the IEEE 802.1Q priority. The device prioritizes a packet by placing the packet in a forwarding queue based on the priority. For example, if a packet's 802.1Q priority is 7, a NetIron stackable Layer 3 Switch uses the high-priority hardware queue to forward the packet. If a packet's 802.1Q priority is 0, the device uses the normal (best-effort) queue.

Software release 07.1.16 enables you to configure a device to use the value in a IP packet's ToS field instead of the 802.1Q priority to forward the packet. The software selects a forwarding queue for the packet based on the priority. For example, you can configure the device to interpret the contents of a packet's ToS field as DSCP. The software reads the value, maps the value into the corresponding priority, then selects a forwarding queue for the packet based on the priority.

In addition, you can enable the device to mark the outbound packet with the DSCP or an 802.1Q priority based on the forwarding priority, so that the next hop for the packet also can forward the packet based on its 802.1Q priority or ToS value (if the device is configured to do so).

ToS-Based QoS Process

When you enable ToS-based QoS on an interface, the feature does the following:

1. Examines the contents of the ToS field.
   - If the trust level is IP precedence, the device interprets the value as an IP precedence value.
   - If the trust level is DSCP, the device interprets the value as a DSCP.
2. Maps the ToS value to a forwarding priority.
   - If the trust level is IP precedence, the device maps the IP precedence to a DSCP, then maps the DSCP to a forwarding priority.
   - If the trust level is DSCP, the device maps the DSCP to a forwarding priority.
3. Places the packet in a forwarding queue based on the forwarding priority.
   - For forwarding priority 0, the device places the packet in the normal (best effort) queue.
   - For forwarding priorities 1 – 7, the device places the packet in the high queue.

4. If marking is enabled, marks the outbound packet.
    - If 802.1Q marking is enabled, the device changes the contents of the outbound packet's 802.1Q priority field to match the ToS-based QoS forwarding priority.
    - If DSCP marking is enabled, the device changes the contents of the outbound packet's ToS field to match the ToS-based QoS DSCP value.

NOTE: If the trust level is DSCP, then the device does not need to mark the packet but instead leaves the DSCP value the same as the value in the inbound packet.

ToS-Based QoS Parameters

To configure a Foundry device to provide QoS based on the ToS field, configure the following parameters:

- Global parameters:
    - Optionally, change the IP precedence to DSCP mapping performed by the device.
    - Optionally, change the DSCP to forwarding priority mapping performed by the device.

  Internally, the device maps the ToS value to a priority for forwarding. You can change the mapping, which can affect the priority the packet receives while being forwarded as well as the forwarding priority or ToS value with which the device marks the outbound packet when it is forwarded.
- Interface parameters:
    - Enable IP ToS-based QoS.
    - Specify the trust level.
    - Optionally, enable CoS marking, DSCP marking, or both. When you enable marking, the device changes the 802.1Q or ToS value or both in the outbound packet based on the results of the QoS priority translations that occur in the device. For example, if you change the DSCP to 802.1Q priority mapping, the device writes the resulting priority in the outbound packet's IEEE frame.

Mapping Parameters

When you enable ToS-based QoS, the device reads the value in an inbound IP packet's ToS field, maps the value to a DSCP value, then maps the DSCP value to a forwarding priority. The device uses the forwarding priority for forwarding the packet within the device.

NOTE: To provide for future enhancements, if you configure the device to interpret the ToS value as an IP precedence, the device maps this value to a DSCP first, then maps the resulting DSCP to a priority for forwarding.

The device uses the following mappings by default.

Table 2.1: Default DSCP to Forwarding Priority Mappings

DSCP value	0 – 7	8 – 15	16 – 23	24 – 31	32 – 41	40 – 47	48 – 55	56 – 63
Forwarding Priority	0	1	2	3	4	5	6	7

Notice that DSCP values range from 0 – 63, whereas forwarding priority values range from 0 – 7. Any DSCP value within a given range is mapped to the same priority. For example, any DSCP value from 8 – 15 is maps to the same priority, 1.

If you configure the device to interpret the value in the ToS field as an IP precedence, the device maps the IP precedence to a DSCP, then maps the resulting DSCP to a forwarding priority. The device uses the following mappings by default.

Table 2.2: Default IP Precedence to DSCP Mappings

IP precedence	0	1	2	3	4	5	6	7
DSCP value	0	8	16	24	32	40	48	56

For example, if you configure the device to interpret the value in the ToS field as an IP precedence, and a packet has the IP precedence value 6, the device maps the value to the lowest value in the DSCP range 48 – 55. The device then maps 48 to the corresponding forwarding priority, in this case 6.

NOTE: When the IP precedence to DSCP mappings and the DSCP to priority mappings are set to their default values as shown in Table 2.1 and Table 2.2, the translation from IP precedence to priority results in the same value. However, if you change either set of mappings, the priority value can differ from the IP precedence value.

Forwarding Queues

After the device maps the inbound packet's 802.1Q or ToS information to a forwarding priority, the device places the packet into a forwarding queue based on the information. Stackable devices map the priority to the following forwarding queues.

Table 2.3: Priority to Forwarding Queue Mappings

Priority	0	1	2	3	4	5	6	7
Forwarding queue	Normal	High	High	High	High	High	High	High

A Stackable device forwards all high priority traffic on a port's outbound queue before forwarding normal priority traffic on the port.

QoS State

To use ToS-based QoS on an interface, you must enable the feature on that interface. Otherwise, the device uses the 802.1Q priority value for forwarding but ignores the ToS value. Moreover, when QoS is disabled on an interface, the interface cannot mark outbound packets. See the "Marking" section below.

Trust Level

The trust level indicates how you want the device to interpret the priority information in all IP version 4 packets received on an interface. You can specify one of the following:

- CoS – The CoS trust level uses the 802.1Q value in the IEEE frame for forwarding. This trust level provides the same prioritization as the QoS service in previous software releases. CoS is the default trust level.
- IP precedence – The IP precedence trust level uses the three most-significant bits in the packet's ToS field as an IP precedence value.
- DSCP – The DSCP trust level uses the six most-significant bits in the packet's ToS field as a DSCP value.

Marking

Marking changes the value in an outbound packet's 802.1Q field or ToS field to match the results of the QoS translations performed by the device.

Marking is disabled by default. You can enable the following types of marking:

- DSCP

- 802.1Q priority (CoS)
- Both DSCP and 802.1Q priority

When you enable marking on a port, the marking applies to packets received by the device through that port.

DSCP marking is applicable only when the trust level is IP precedence; 802.1Q priority marking is applicable only when the trust level is IP precedence or DSCP. Neither type of marking is applicable if the trust level is CoS, since the CoS trust level does not examine the contents of the ToS field at all.

NOTE: DSCP marking does not apply when the trust level is CoS because this trust level does not apply to the ToS field. DSCP marking does not apply when the trust level is DSCP, because in this case the ToS field already contains a DSCP value.

Configuring ToS-Based QoS

If you plan to use the default IP precedence to DSCP mappings and the default DSCP to 802.1Q priority mappings, then you do not need to configure any global parameters for ToS-based QoS. To enable ToS-based QoS, use the following procedures.

Enabling ToS-Based QoS

To enable ToS-based QoS on an interface, use the following CLI method. You must enable the feature before you can specify the trust level or enable marking.

USING THE CLI

To enable ToS-based QoS on port 1, enter the following commands:

```
NetIron(config-if-1)# qos-tos
```

Syntax: [no] qos-tos

Specifying the Trust Level

The trust level specifies where you want the device to get the QoS value for an IP version 4 packet received on the interface. To use ToS-based QoS for packets received on the interface, you must enable the IP precedence or DSCP trust level.

To configure the trust level, use the following CLI method.

USING THE CLI

To set the trust level for port 1 to DSCP, enter the following commands:

```
NetIron(config)# interface ethernet 1
NetIron(config-if-1)# qos trust dscp
```

Syntax: [no] qos trust cos | ip-prec | dscp

The **cos | ip-prec | dscp** parameter specifies the trust level.

- **cos** – The device uses the 802.1Q priority value in the packet's IEEE frame header. This is the default.
- **ip-prec** – The device uses the three most-significant bits in the packet's ToS field and interprets them as an IP precedence value.
- **dscp** – The device uses the six most-significant bits in the packet's ToS field and interprets them as a DSCP value.

NOTE: If you specify **cos**, the device does not examine the contents of the ToS field and therefore does not perform ToS-based QoS.

Enabling Marking

Marking changes the value of an outbound packet's 802.1Q priority field or ToS field to match the results of the QoS mappings performed by the Foundry device. When you enable marking on a port, the marking applies to packets that enter the device through that port. To enable marking, use the following CLI method.

USING THE CLI

To enable CoS marking for port 1, enter the following command:

```
NetIron(config-if-1)# qos mark cos
```

Syntax: [no] qos mark cos | dscp

The **cos | dscp** parameter specifies the type of marking.

- **cos** – The device changes the outbound packet's 802.1Q priority value to match the results of the device's QoS mapping from the ToS value (IP precedence or DSCP) into the 802.1Q value.
- **dscp** – The device changes the outbound packet's ToS value to match the results of the device's QoS mapping from IP precedence to DSCP.

NOTE: DSCP marking does not apply when the trust level is CoS because this trust level does not apply to the ToS field. DSCP marking does not apply when the trust level is DSCP, because in this case the ToS field already contains a DSCP value.

Changing the Mappings

When you specify the IP precedence trust level or the DSCP trust level and you enable ToS-based QoS, the Foundry device maps the ToS value into a forwarding priority. By default, the device uses the mappings shown in Table 2.1 and Table 2.2 in "Mapping Parameters" on page 2-29.

You can globally change the DSCP to forwarding priority mappings or the IP precedence to DSCP mappings. The device forwards the packet based on the changed mappings. In addition, if you enable marking, the device changes the outbound packet's 802.1Q priority or ToS value based on the changed mappings.

Changing the DSCP to Forwarding Priority Mappings

To change the DSCP to forwarding priority mappings, use the following CLI method.

USING THE CLI

To change the DSCP to forwarding priority mappings for all the DSCP ranges, enter commands such as the following at the global CONFIG level of the CLI:

```
BigIron(config)# qos map dscp-priority 0 2 3 4 to 1
BigIron(config)# qos map dscp-priority 8 to 5
BigIron(config)# qos map dscp-priority 16 to 4
BigIron(config)# qos map dscp-priority 24 to 2
BigIron(config)# qos map dscp-priority 32 to 0
BigIron(config)# qos map dscp-priority 40 to 7
BigIron(config)# qos map dscp-priority 48 to 3
BigIron(config)# qos map dscp-priority 56 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.


```
BigIron(config-if-1)# show qos
```

...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
```

For information about the rest of this display, see “Displaying Configuration Information” on page 2-34.

Syntax: [no] qos 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 forwarding priority.

Changing the IP Precedence to DSCP Mappings

To change the IP precedence to DSCP mappings, use the following CLI method.

USING THE CLI

To change the IP precedence to DSCP mappings, enter a command such as the following at the global CONFIG level of the CLI:

```
NetIron(config)# qos map ip-prec-dscp 0 32 24 48 16 8 56 40
```

These commands configure the mappings displayed in the IP precedence to DSCP portion of the QoS information display.

```
BigIron(config-if-1)# show qos
```

...portions of table omitted for simplicity...

```
IP Precedence-DSCP map:

  ip-prec:     0     1     2     3     4     5     6     7
  ------------------------------------------------------------
     dscp:     0    32    24    48    16     8    56    40
```

For information about the rest of this display, see “Displaying Configuration Information” .

Syntax: [no] qos map ip-prec-dscp <dscp1> <dscp2> <dscp3> <dscp4> <dscp5> <dscp6> <dscp7> <dscp8>

The <dscp1> ... <dscp8> parameters specify the DSCP values you are mapping to the IP precedence values. You must enter DSCP values for all eight IP precedence values, in order from IP precedence value 0 – 7.

Displaying Configuration Information

To display configuration information for ToS-based QoS, use the following CLI method.

USING THE CLI

To display configuration information, enter the following command at any level of the CLI:

```
BigIron(config-if-1)# show qos
Interface QoS , Marking and Trust Level:
  i/f  |    QoS    |   Mark    |  Trust-Level
-------+-----------+-----------+---------------
1      |    Yes    |    COS    |   IP Prec
2      |    No     |    No     |   Layer 2 CoS
3      |    No     |    No     |   Layer 2 CoS
4      |    No     |    No     |   Layer 2 CoS
5      |    Yes    |    No     |   DSCP
6      |    No     |    No     |   Layer 2 CoS
7      |    No     |    No     |   Layer 2 CoS
8      |    No     |    No     |   Layer 2 CoS
9      |    No     |    No     |   Layer 2 CoS
10     |    No     |    No     |   Layer 2 CoS
11     |    No     |    No     |   Layer 2 CoS
12     |    No     |    No     |   Layer 2 CoS
13     |    No     |    No     |   Layer 2 CoS
14     |    No     |    No     |   Layer 2 CoS
15     |    No     |    No     |   Layer 2 CoS
16     |    No     |    No     |   Layer 2 CoS
17     |    No     |    No     |   Layer 2 CoS
18     |    No     |    No     |   Layer 2 CoS
ve1    |    Yes    | COS, DSCP|   IP Prec
ve10   |    No     |    No     |   Layer 2 CoS

IP Precedence-DSCP map:

  ip-prec:     0     1     2     3     4     5     6     7
  ------------------------------------------------------------
     dscp:    10    11    12    13    14    15    16    17


DSCP-Priority map: (dscp = d1d2)

     d2|  0   1   2   3   4   5   6   7   8   9
  d1   |
  -----+----------------------------------------
   0   |  0   0   0   0   0   0   0   0   1   1
   1   |  6   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
```

Syntax: show qos

This command shows the following information.

Table 2.4: ToS-Based QoS Configuration Information

This Field...	Displays...
Interface QoS, Marking and Trust Level information	
if	The interface
QoS	The state of ToS-based QoS on the interface. The state can be one of the following: • No – Disabled • Yes – Enabled
Mark	The marking type enabled on the interface. The marking type can be any of the following: • COS – CoS marking is enabled. • DSCP – DSCP marking is enabled. • No – Marking is not enabled.
Trust-Level	The trust level enabled on the interface. The trust level can be one of the following: • DSCP • IP Prec • Layer 2 CoS
IP Precedence-DSCP map	
ip-prec and dscp	The IP precedence to DSCP 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.
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

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. To do so, use either of the following methods.

USING THE CLI

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.

```
BigIron(config)# relative-utilization 1 uplink eth 1/1 downlink eth 1/2 to 1/3
BigIron(config)# write memory
```

Syntax: [no] relative-utilization <num> uplink ethernet <portnum> [to <portnum> | <portnum>…] downlink ethernet <portnum> [to <portnum> | <portnum>…]

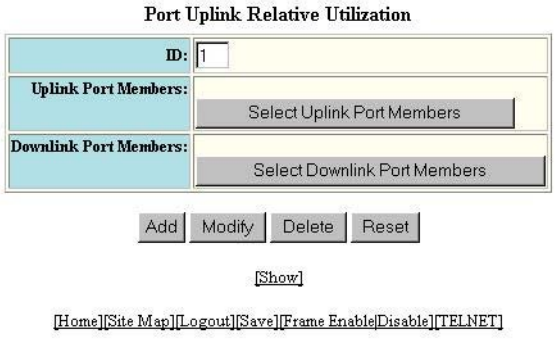
The <num> parameter specifies the list number. You can configure up to four lists. Specify a number from 1 – 4.

The **uplink ethernet** parameters and the port number(s) you specify after the parameters indicate the uplink port(s).

The **downlink ethernet** parameters and the port number(s) you specify after the parameters indicate the downlink port(s).

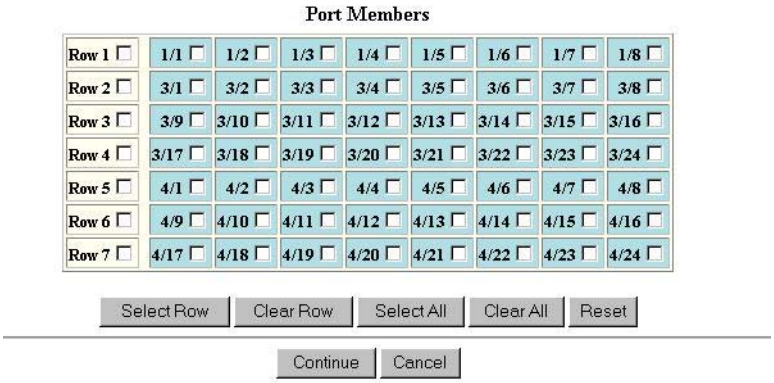
USING THE WEB MANAGEMENT INTERFACE

1. Log on to the device using a valid user name and password for read-write access. The System configuration dialog is displayed.
2. Click on the plus sign next to Configure in the tree view to expand the list of configuration options.
3. Click on the plus sign next to Port in the tree view to display the configuration options.
4. Select the link to the port type you want (for example, Ethernet) to display the Port table.
5. Click on the Relative Utilization link at the top of the panel to display the Port Uplink Relative Utilization panel, as shown in the following example:



6. Enter the ID for the link utilization list in the ID field. You can specify a number from 1 – 4.

7. Click the Select Uplink Port Members button. A Port Members panel similar to the following is displayed.



8. Select the boxes next to the ports you want to include in the uplink list. When you have finished, click Continue.
9. On the Port Uplink Relative Utilization panel, click the Select Downlink Port Members button to display a Port Members panel for downlink ports.
10. Select the boxes next to the ports you want to include in the downlink list. When you have finished, click Continue.
11. On the Port Uplink Relative Utilization panel, click the Add button create the uplink utilization list.
12. Select the Save link at the bottom of the dialog, then select Yes when prompted to save the configuration change to the startup-config file on the device's flash memory.

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 uplink utilization percentages, use either of the following methods.

USING THE CLI

To display an uplink utilization list, enter a command such as the following at any level of the CLI:

```
BigIron(config)# 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.

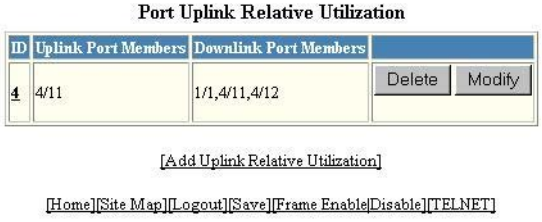
```
BigIron(config)# show relative-utilization 1
uplink: ethe 1
30-sec total uplink packet count = 3011
packet count ratio (%)
  1/ 2:100   1/ 3:100
```

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.

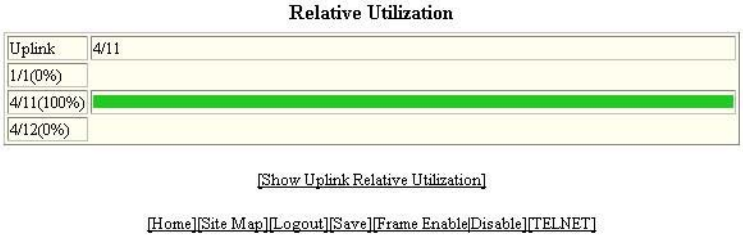
```
BigIron(config)# show relative-utilization 1
uplink: ethe 1
30-sec total uplink packet count = 2996
packet count ratio (%)
  1 /2:100   1/ 3:---
```

USING THE WEB MANAGEMENT INTERFACE

1. Log on to the device using a valid user name and password for read-write access. The System configuration dialog is displayed.
2. Click on the plus sign next to Configure in the tree view to expand the list of configuration options.
3. Click on the plus sign next to Port in the tree view to display the configuration options.
4. Select the link to the port type you want (for example, Ethernet) to display the Port table.
5. Click on the Relative Utilization link at the top of the panel to display the Port Uplink Relative Utilization panel.
6. Click on the Show link. A panel listing the configured uplink utilization lists is displayed:



7. Click on the ID of an uplink utilization list to display utilization percentages for the ports in the list.



This panel displays a graph of the percentage of the uplink's bandwidth that each of the downlink ports used during the most recent 30-second port statistics interval.

Chapter 3

Configuring Quality of Service on a FastIron Edge Switch and FastIron Edge Switch X-Series

This chapter describes how to configure Quality of Service (QoS) on a FastIron Edge Switch (FES) or a FastIron Edge Switch X-Series.

See the appropriate section, as follows:

- "QoS on a FastIron Edge Device" on page 3-1
- "QoS on a FastIron Edge Switch X-Series" on page 3-8

NOTE: The QoS and ToS-based QoS features described in this chapter apply only to the FastIron Edge Switch and FastIron Edge Switch X-Series. To configure QoS on other devices, see "Configuring Basic Quality of Service" on page 2-1. To configure ToS-based QoS on JetCore devices, 10 Gigabit Ethernet modules, and VM1 modules, see "Configuring Enhanced Quality of Service" on page 4-1.

QoS on a FastIron Edge Device

The FastIron Edge Switch provides the following QoS features:

- Configurable queuing mechanisms
- Automatic mapping of 802.1p priorities to QoS queues
- 802.1Q support for features such as Voice over IP (VoIP)
- Configurable reassignment of traffic from one queue to another
- Prioritization through mapping of DSCP values to hardware forwarding queues (ToS-based QoS)

The Queues

The FastIron Edge Switch has the following queues:

Table 3.1: QoS Queues

Queue	Description
qosp3	The highest priority queue. This queue corresponds to 802.1p prioritization levels 6 and 7 and Foundry priority levels 6 and 7.

Table 3.1: QoS Queues (Continued)

Queue	Description
qosp2	The second-highest priority queue. This queue corresponds to 802.1p prioritization levels 4 and 5 and Foundry priority levels 4 and 5.
qosp1	The third-highest priority queue. This queue corresponds to 802.1p prioritization levels 2 and 3 and Foundry priority levels 2 and 3.
qosp0	The lowest priority queue. This queue corresponds to 802.1p prioritization levels 0 and 1 and Foundry priority levels 0 and 1.

The queue names listed above are the default names. If desired, you can rename the queues as described in "Renaming the Queues" on page 3-3".

You can classify packets and assign them to specific queues based on the following criteria:

- Incoming port (also called ingress port)
- Layer 3 and Layer 4 information (IP and TCP/UDP source and destination information)
- Static MAC entry
- AppleTalk socket number

See "Assigning QoS Priorities to Traffic" on page 3-5.

In addition, in release 02.0.00 and later, the device automatically maps a packet's DSCP value to a hardware forwarding queue. See "Type of Service (ToS) Based QoS" on page 3-5.

Queuing Methods

In sofware release 02.0.00 and later, you can configure a FastIron Edge Switch to use one of the following queuing methods:

- Weighted Round Robin – A weighted fair queuing algorithm is used to rotate service among the four queues. The rotation is based on the weights you assign to each queue. This is the default queuing method and uses a default set of queue weights. This method rotates service among the four queues, forwarding a specific number of packets in one queue before moving on to the next one.

  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.

  In release 02.0.00 and later, Weighted Round Robin is the default.

  NOTE: The weighted round robin method is not supported in releases prior to 02.0.00 and later.

- Strict Priority – 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.

Selecting the Queuing Method

The FastIron Edge Switch uses the weighted fair queuing method of packet prioritization by default. To change the method to strict priority or back to weighted fair queuing, use one of the following methods.

USING THE CLI

To change the queuing method from weighted round robin to strict queuing, enter the following commands:

```
FES4802 Router(config)# qos mechanism strict
```


To change the method back to weighted round robin, enter the following command:

```
FES4802 Router(config)# qos mechanism weighted
```

Syntax: [no] qos mechanism strict | weighted

Configuring the Queues

Each of the four 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 are qosp3, qosp2, qosp1, and qosp0. You can change one or more of the names if desired. To do so, use one of the following methods.

To rename queue qosp3 (the premium queue) to "92-octane", enter the following commands:

```
FES4802 Router(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 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 four QoS queues receive the following minimum guaranteed percentages of a port's total bandwidth.

Queue	Default Minimum Percentage of Bandwidth
qosp3	75%
qosp2	15%
qosp1	5%
qosp0	5%

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.

The bandwidth allocated to each queue is based on the relative weights of the queues. Release 02.0.00 enables you to change the bandwidth percentages allocated to the queues by changing the weights of the queues.

There is no minimum bandwidth requirement for a give queue. For example, queue qosp3 is not required to have at least 50% of the bandwidth.

To change the bandwidth percentages for the queues, enter a command such as the following. Note that this example uses the default queue names.

```
FES4802 Router(config)# qos profile qosp3 65 qosp2 20 qosp1 8 qosp0 7
Profile qosp3     : PREMIUM     bandwidth requested  65% calculated  65%
Profile qosp2     : HIGH        bandwidth requested  20% calculated  20%
Profile qosp1     : NORMAL      bandwidth requested  8% calculated   8%
```



```
Profile qosp0     : BEST-EFFORT bandwidth requested   7% calculated   7%
```

Syntax: [no] qos profile <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.

NOTE: The percentages you enter must equal 100.

NOTE: The FastIron Edge Switch does 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. The FastIron Edge Switch queues do not have a minimum required bandwidth percentage, so adjustment is unnecessary. For example, queue qosp3 on a BigIron device must have at least 50% of the bandwidth. There is no such requirement on the FastIron Edge Switch.

802.1p Support

The FastIron Edge Switch maps the 802.1p priority value of each packet to one of the device's four QoS queues. The following table shows the default incoming mapping.

Priority Level	Queue
6, 7	qosp3
4, 5	qosp2
2, 3	qosp1
0, 1	qosp0

By default, all traffic has priority 0. You can assign a higher priority to traffic based on the following:

- Incoming port (sometimes called the ingress port)
- Static MAC entry
- Layer 3 and Layer 4 information (IP and TCP/UDP source and destination information)
- AppleTalk socket

See "Assigning QoS Priorities to Traffic" on page 3-5.

In addition, in release 02.0.00 and later the device automatically maps a packet's DSCP value to a hardware forwarding queue. See "Type of Service (ToS) Based QoS" on page 3-5.

802.1Q Marking

If a packet enters the device on a tagged port, the device prioritizes the packet by mapping its 802.1Q value to a hardware forwarding queue. If a packet enters the device on an untagged port but is forwarded on a tagged port, the device tags the packet and adds an 802.1Q value. The 802.1Q value is based on the priority assigned to the packet as it travels through the device.

The following table shows the default outgoing mapping.

Queue	VLAN Priority Tag
qosp3	6
qosp2	4
qosp1	2
qosp0	0

Assigning QoS Priorities to Traffic

All traffic is in the best-effort queue (qosp0) by default. You can assign traffic to a higher queue based on the following:

- Incoming port (sometimes called the ingress port)
- Static MAC entry
- Layer 3 and Layer 4 information (IP and TCP/UDP source and destination information)
- AppleTalk socket

Changing a Port's Priority

To change the QoS priority of port 1 to the premium queue (qosp3), enter the following commands:

```
FES4802 Router(config)# interface ethernet 1
FES4802 Router(config-if-1)# priority 7
```

The device will change the 802.1p priority of traffic received on port 1 to priority 7.

Syntax: [no] priority <num>

The <num> parameter can be from 0 – 7 and specifies the IEEE 802.1 equivalent to one of the four QoS queues.

Viewing QoS Settings

To display the QoS settings for all the queues, enter the following command:

```
FES4802 Router(config)# show qos-profiles all
bandwidth scheduling mechanism: weighted priority
Profile qosp3     : PREMIUM     bandwidth requested  65% calculated  65%
Profile qosp2     : HIGH        bandwidth requested  20% calculated  20%
Profile qosp1     : NORMAL      bandwidth requested  8% calculated   8%
Profile qosp0     : BEST-EFFORT bandwidth requested   7% calculated   7%
```

Syntax: show qos-profiles all | <name>

The **all** parameter displays the settings for all four queues. The <name> parameter displays the settings for the specified queue.

Type of Service (ToS) Based QoS

The FastIron Edge Switch supports basic ToS-based QoS. Software releases 03.0.00 and later also support marking of the DSCP value. However, it does not support other advanced ToS-based QoS features as described in the "Configuring Enhanced Quality of Service" chapter.

In software release 02.0.00 and later, the FastIron Edge Switch 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 four hardware forwarding queues (qosp0, qosp1, qosp2, or qosp3). During a forwarding cycle, the device gives more preference to the higher numbered queues, so that more packets are forwarded from these queues. Queue qosp3 receives the highest preference while queue qosp0, the best-effort queue, receives the lowest preference.

Type-of-Service (ToS) based QoS is enabled by default.

Enabling ToS-Based QoS

Basic ToS-Based QoS is enabled on the FastIron Edge Switch by default. To re-enable the feature after it has been disabled, enter the following command at the global CONFIG level of the CLI:

```
FES4802 Router(config)# port-priority
```

Changing 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 3.2 list the default mappings of DSCP values to internal forwarding priority values.

Table 3.2: Default DSCP to Internal Forwarding Priority Mappings

DSCP value	0 – 7	8 – 15	16 – 23	24 – 31	32 – 41	40 – 47	48 – 55	56 – 63
Internal Forwarding Priority	0	1	2	3	4	5	6	7

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.

- qosp3 – the highest priority queue
- qosp2 – the second-highest priority queue
- qosp1 – the third-highest priority queue
- qosp0 – the best-effort (lowest priority) queue

Table 3.3 list the default mappings of internal forwarding priority values to the hardware forwarding queues.

Table 3.3: Default Internal Forwarding Priority to Hardware Forwarding Queue Mappings

Internal Forwarding Priority	0	1	2	3	4	5	6	7
Forwarding Queue	qosp0	qosp0	qosp1	qosp1	qosp2	qosp2	qosp3	qosp3

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:

```
FES4802 Router(config)# qos-tos map dscp-priority 0 2 3 4 to 1
FES4802 Router(config)# qos-tos map dscp-priority 8 to 5
FES4802 Router(config)# qos-tos map dscp-priority 16 to 4
FES4802 Router(config)# qos-tos map dscp-priority 24 to 2
FES4802 Router(config)# qos-tos map dscp-priority 32 to 0
FES4802 Router(config)# qos-tos map dscp-priority 40 to 7
FES4802 Router(config)# qos-tos map dscp-priority 48 to 3
FES4802 Router(config)# qos-tos map dscp-priority 56 to 6
FES4802 Router(config)# ip rebind-acl all
```

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.

```
FES4802 Router(config-if-1)# 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:

```
BigIron(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:

- qosp3

- qosp2
- qosp1
- qosp0

Viewing ToS-Based QoS Settings

To display configuration information for ToS-based QoS, enter the following command at any level of the CLI.

Syntax: show qos-tos

QoS on a FastIron Edge Switch X-Series

The FastIron Edge Switch X-Series provides the following QoS features:

- Configurable queuing mechanisms
- Automatic mapping of 802.1p priorities to QoS queues
- 802.1Q support for features such as Voice over IP (VoIP)
- Configurable reassignment of traffic from one queue to another
- Prioritization through mapping of DSCP values to hardware forwarding queues (ToS-based QoS)

The Queues

The FES X-Series provides the following eight queues.

Queue	Description
qosp7	Priority 7 (highest-priority queue)
qosp6	Priority 6
qosp5	Priority 5
qosp4	Priority 4
qosp3	Priority 3
qosp2	Priority 2
qosp1	Priority 1
qosp0	Priority 0 (lowest-priority queue)

The queue names listed above are the default names. If desired, you can rename the queues as described in “Renaming the Queues” on page 3-9.

Queuing Methods

The FES X-Series supports the following queueing mechanisms:

- Strict Priority (SP)
- Weighted Round Robin (WRR)

The default is WRR.

Strict Priority (SP)

SP ensures service for high priority traffic. To do so, SP services all packets in the high priority queues before moving to the lower priority queues. SP processes packets in qosp7 before processing any packets in qosp6, then processes packets in qosp6 before processing any packets in qosp5, and so on.

Weighted Round Robin (WRR)

WRR ensures that all eight queues are serviced during each cycle. WRR rotates service among all eight queues, based on the weights assigned to each queue. WRR forwards a specific number of bytes in one queue before moving on to the next one in a round-robin fashion. This process avoids starvation of the queues.

NOTE: The FES X-Series' queue cycles are based on bytes. The device services a given number of bytes (based on the weight) in each queue cycle. BigIron queue cycles are based on packets.

The bytes-based scheme is more accurate compared to a packets-based scheme if there is a large variation in the size of the packets.

Selecting the Queuing Method

The FastIron Edge Switch X-Series uses the weighted fair queuing method of packet prioritization by default. To change the method to strict priority or back to weighted fair queuing, use one of the following methods.

USING THE CLI

To change the queuing method from weighted round robin to strict queuing, enter the following commands:

```
FESX424 Router(config)# qos mechanism strict
```

To change the method back to weighted round robin, enter the following commands:

```
FESX424 Router(config)# qos mechanism weighted
```

Syntax: [no] qos mechanism strict | weighted

Configuring the Queues

Each of the eight queues has the following configurable parameters:

- The queue name
- The minimum bandwidth percentages for the queues

Renaming the Queues

The default queue names are qosp7, qosp6, qosp5, qosp4, qosp3, qosp2, qosp1, and qosp0. You can change one or more of the names if desired.

To rename queue qosp7 (the premium queue) to "91-octane", enter the following command:

```
FESX424 Switch(config)# qos name qosp7 91-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 Bandwidth Allocations

To change the bandwidth percentages for the queues, enter commands such as the following. Note that this example uses the default queue names.

```
FESX424 Switch(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. The FES X-Series 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 total of the percentages you enter must equal 100.

NOTE: The FES X-Series does 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.

NOTE: When sFlow is enabled, the FES X-Series supports seven priorities instead of eight. When sFlow is enabled, Priority 1 is not used. Any values assigned to queue 1 will be directed to queue 0.

802.1p Support

The FES X-Series maps the 802.1p priority value of each packet to one of the device's eight QoS queues. The following table shows the default incoming mapping.

Priority Level	Queue
7	qosp7
6	qosp6
5	qosp5
4	qosp4
3	qosp3
2	qosp2
1	qosp1

Priority Level	Queue
0	qosp0

By default, all traffic has priority 0. You can assign a higher priority to traffic based on the following:

- Incoming port (sometimes called the ingress port)
- Static MAC entry
- Layer 3 and Layer 4 information (IP and TCP/UDP source and destination information)

In addition, the device automatically maps a packet's DSCP value to a hardware forwarding queue. See "Type of Service (ToS) Based QoS" on page 3-12.

802.1Q Marking

If a packet enters the device on a tagged port, the device prioritizes the packet by mapping its 802.1Q value to a hardware forwarding queue. If a packet enters the device on an untagged port but is forwarded on a tagged port, the device tags the packet and adds an 802.1Q value. The 802.1Q value is based on the priority assigned to the packet as it travels through the device.

The following table shows the default outgoing mapping.

Queue	VLAN Priority Tag
qosp7	7
qosp6	6
qosp5	5
qosp4	4
qosp3	3
qosp2	2
qosp1	1
qosp0	0

Assigning QoS Priorities to Traffic

All traffic is in the lowest-priority queue (qosp0) by default. You can assign traffic to a higher queue based on the following:

- Incoming port (sometimes called the ingress port)
- Static MAC entry
- Layer 3 and Layer 4 information (IP and TCP/UDP source and destination information)

Changing a Port's Priority

To change the QoS priority of port 1 to the highest-priority queue (qosp7), enter the following commands:

```
FESX424 Switch(config)# interface ethernet 1
FESX424 Switch(config-if-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 the eight QoS queues.

Viewing QoS Settings

To display the QoS settings for all the queues, enter the following command:

```
FESX424 Switch(config)# 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.

Type of Service (ToS) Based QoS

The FES X-Series supports basic ToS-based QoS. Basic ToS-based QoS provides prioritization to a packet being forwarded out the device, by mapping the packet's DSCP value to an internal forwarding priority, which is mapped to a hardware forwarding queue.

The FES X-Series also supports marking of the DSCP value. However, it does not support other advanced ToS-based QoS features described in the "JetCore Type of Service (ToS) Based QoS" chapter of the *Foundry Enterprise Configuration and Management Guide*.

Chapter 4
Configuring Enhanced Quality of Service

This chapter applies to the following devices:

- 10 Gigabit Ethernet modules
- JetCore devices
- VM1 modules

Foundry devices can read Layer 2 and Layer 3 Quality of Service (QoS) information in an IP packet and select a forwarding queue for the packet based on the information. In addition, Foundry devices also can modify the packet's QoS information so that the packet's next hop uses the modified information.

By default, Type-of-Service (ToS) based QoS is disabled.

Basic and Advanced ToS-Based QoS

IP ToS-based QoS offers two levels of support, basic and advanced.

- Basic – When you globally enable ToS-based QoS, the Foundry device maps a packet's DiffServ Control Point (DSCP) value to an internal forwarding priority, and sends the packet to the hardware forwarding queue that corresponds to the internal forwarding priority.

  If the egress port for the packet is an 802.1q tagged port, the device also maps the device's DSCP value to an 802.1p value and changes the packet's 802.1p value accordingly.

- Advanced – In addition to globally enabling basic support, you also can enable advanced ToS-based QoS on individual interfaces. Enabling advanced ToS-based QoS on an interface allows you to specify the trust level and packet marking used for packets received on that interface. The ***trust level*** determines the type of QoS information the device uses for performing QoS. ***Marking*** is the process of changing the packet's QoS information for the next hop. Trust levels and marking are described in detail in "Classification, Marking, and Scheduling" on page 4-2.

NOTE: You cannot use advanced ToS-based QoS and rate limiting on the same interface.

Basic and advanced ToS-based QoS each map a packet's QoS value to an internal forwarding priority. The internal forwarding priorities are mapped to one of the four hardware forwarding queues (qosp0, qosp1, qosp2, or qosp3). During a forwarding cycle, the device gives more preference to the higher numbered queues, so that more packets are forwarded from these queues. Queue qosp3 receives the highest preference while queue qosp0, the best-effort queue, receives the lowest preference.

NOTE: For finer control of QoS, use the QoS options for ACLs. Refer to “QoS Options for IP ACLs (Rule-Based ACLs)” on page 6-44.

QoS Support When IP ToS-Based QoS Is Disabled

When ToS-based QoS is disabled, a packet’s priority can be changed only by directly resetting its internal forwarding priority as it travels through the system or by using an ACL to explicitly change the priority. (See “Alternative QoS Methods” on page 4-5.) Otherwise, the packet’s Layer 2 and Layer 3 QoS information is not examined by the device and does not affect how the packet is forwarded through the device.

Classification, Marking, and Scheduling

The ToS-based QoS process involves the following stages:

- Classification
- Marking
- Scheduling

Classification

Classification is the process of selecting packets on which to perform QoS and reading the QoS information. A packet can have multiple types of QoS information. The ***trust level*** in effect on an interface determines the type of QoS information the device uses for performing QoS. The trust level can be one of the following:

- Layer 2 CoS – The 802.1p priority in the Ethernet frame. The priority is a value from 0 – 7. The 802.1p priority is also called the ***Class of Service (CoS)***.

  NOTE: This trust level is not supported on 10 Gigabit Ethernet modules.

- Layer 3 IP Precedence – The value in the three most significant bits of the IP packet header’s 8-bit ToS field. The IP Precedence is a value from 0 – 7. The IP Precedence values are described in RFC 791.
- Layer 3 DSCP – The value in the six most significant bits of the IP packet header’s 8-bit ToS field. The DSCP is the six most significant bits in the IP packet header’s ToS field. The DSCP value is sometimes called the ***DiffServ*** value, and can be from 0 – 63. The DSCP values are described in RFCs 2474 and 2475.

NOTE: Basic ToS-based QoS uses the DSCP trust level and 802.1p marking (see below) by default. For other trust levels and marking for an interface, you must enable advanced ToS-based QoS on the interface. See “Enabling Basic ToS-Based QoS” on page 4-7.

Marking

Marking is the process of changing the packet’s QoS information 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. When marking is disabled, 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.

NOTE: Starting in software release 07.6.03, the VM1 supports marking of ToS bits. This is the only enhanced-QoS feature supported on VM1. VM1 does not support basic ToS-based QoS. Also, although the VM1 uses advanced ToS-based QoS, it does not support ToS-based QoS scheduling.

NOTE: Because of hardware limitations on JetCore modules, if the outbound interface for a packet is an 802.1q interface (the interface is tagged), the device may change the 802.1p priority of the packet to one value lower, even if you have not configured marking of this value. This can occur if the packet's 802.1 priority is odd (1, 3, 5, or 7). In this case, the device changes the 802.1p priority to the next lower value: 7 is changed to 6, 5 is changed to 4, 3 is changed to 2, and 1 is changed to 0. If the packet's 802.1p priority is an even value, the value is unchanged. This behavior does not affect the DSCP/ToS value.

Scheduling

Scheduling is the process of mapping a packet to an internal forwarding queue based on its QoS information, and placing the packet into one of the four hardware forwarding queues for forwarding. The Foundry device maps the packet's QoS value into a CoS or DSCP value, then maps that value to an internal forwarding queue. The device then places the packet in the hardware forwarding queue corresponding to the internal forwarding queue.

You can modify the scheduling by changing the following mappings:

- CoS –> DSCP
- IP Precedence –> DSCP
- DSCP –> DSCP
- DSCP –> internal forwarding priority

The first three mappings are the same ones described in "Classification" and are applicable for DSCP marking. The trusted QoS source (CoS, IP Precedence, or DSCP) is mapped to a DSCP value and the packet is marked with that DSCP value.

The DSCP –> internal forwarding priority mapping is used to translate the results of the first three mappings into a value that the Foundry device can use to select a hardware forwarding queue. In addition, if the outgoing interface is an 802.1q tagged interface, the DSCP value is mapped to an 802.1p value and the packet is marked with the 802.1p value.

NOTE: VM1 does not support ToS-based QoS scheduling for IPv4 traffic

NOTE: In the current release, the device schedules a packet by mapping the higher of the packet's 802.1p or DSCP/ToS values to one of the hardware forwarding queues. Unless other priority settings change the packet to a higher queue, the queue sleeted when the packet is received is used for forwarding the packet. A packet's forwarding priority (hardware forwarding queue) can be changed to a higher queue but cannot be changed to a lower queue.

Default QoS Mappings

The Foundry device can map an incoming packet's CoS (802.1p), IP Precedence, or DSCP value into a DSCP value, and can map the DSCP value to an internal forwarding priority. The mappings are used for forwarding the packet to an output queue within the Foundry device and for marking the packet.

The following tables list the default mappings. You can change the mappings if needed. See "Changing the QoS Mappings" on page 4-8.

Default CoS –> DSCP Mappings

Table 4.1 list the default mappings of CoS (802.1p) values to DSCP values. These mappings are used if the trust level is CoS and DSCP marking is enabled.

Table 4.1: Default 802.1p to DSCP Mappings

802.1p	0	1	2	3	4	5	6	7
DSCP value	0	8	16	24	32	40	48	56

Default IP Precedence –> DSCP Mappings

Table 4.2 list the default mappings of IP precedence values to DSCP values. These mappings are used if the trust level is IP Precedence and DSCP marking is enabled.

Table 4.2: Default IP Precedence to DSCP Mappings

IP precedence	0	1	2	3	4	5	6	7
DSCP value	0	8	16	24	32	40	48	56

The device maps the IP precedence value to the lowest value in the DSCP range.

Default DSCP –> DSCP Mappings

By default, the device maps a packet's DSCP value to the same DSCP value. For example, if the packet has DSCP value 63 when the packet is received, the packet still has DSCP value 63 when the packet is placed in the hardware forwarding queue.

Default DSCP –> Internal Forwarding Priority Mappings

Table 4.3 list the default mappings of DSCP values to internal forwarding priority values.

Table 4.3: Default DSCP to Internal Forwarding Priority Mappings

DSCP value	0 – 7	8 – 15	16 – 23	24 – 31	32 – 41	40 – 47	48 – 55	56 – 63
Internal Forwarding Priority	0	1	2	3	4	5	6	7

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.

- qosp3 – the highest priority queue
- qosp2 – the second-highest priority queue
- qosp1 – the third-highest priority queue
- qosp0 – the best-effort (lowest priority) queue

Table 4.4 list the default mappings of internal forwarding priority values to the hardware forwarding queues.

Table 4.4: Default Internal Forwarding Priority to Hardware Forwarding Queue Mappings

Internal Forwarding Priority	0	1	2	3	4	5	6	7
Forwarding Queue	qosp0	qosp0	qosp1	qosp1	qosp2	qosp2	qosp3	qosp3

Layer 4 CAM Usage

Basic ToS-based QoS does not use Layer 4 CAM entries. Advanced ToS-based QoS does use Layer 4 CAM entries of the interface where the feature is enabled. The number of CAM entries used by QoS depends on the trust level, as listed in Table 4.5.

Table 4.5: Layer 4 CAM Usage

Trust level	**Number of Layer 4 CAM entries**
CoS	4 per interface
IP Precedence	7 per interface
DSCP	63 per interface

Since advanced ToS-based QoS uses Layer 4 CAM entries, Foundry recommends that you enable advanced ToS-based QoS on an interface only if required by the type of traffic received on the interface. Other features including ACLs, PBR, and NAT also require Layer 4 CAM entries.

Using ACLs, PBR, or NAT and IP ToS-Based QoS

You can use ACLs and IP ToS-based QoS on the same interfaces. However, for basic and advanced QoS, if an interface has an ACL applied to it, the only packets on that interface that are eligible for IP ToS-based QoS are the packets that match the **permit ip any any** ACL. A packet that matches any other ACL on the interface is not eligible for IP ToS-based QoS. This is true regardless of whether the ACLs are used for traffic filtering, for PBR, or for NAT. Nonetheless, you still can provide QoS for these packets using the ACL options listed in “Alternative QoS Methods” on page 4-5.

DSCP Processing for Traffic Forwarded by the CPU

In general, most traffic on a Layer 2 Switch or Layer 3 Switch is forwarded in hardware. However, some traffic, including traffic forwarded on interfaces that have outbound ACLs, is sent to the CPU for forwarding. On a Layer 2 Switch, DSCP mapping (basic ToS-based QoS) is not performed on traffic forwarded by the CPU. DSCP mapping is performed for traffic forwarded by the CPU on a Layer 3 Switch.

Alternative QoS Methods

If you do not want to enable IP ToS-based QoS, you still can configure the device to prioritize and even mark packets. When ToS-based QoS is disabled, Foundry devices prioritize IP packets as follows:

- If the packet's internal forwarding priority is reset directly, the reset priority is used. You can directly reset a packet's internal forwarding priority based on any of the following:
  - Incoming port (sometimes called ingress port)
  - IP source and destination addresses
  - Layer 4 source and destination information (for all IP addresses or specific IP addresses)
  - Static MAC entry
  - AppleTalk socket number
  - Layer 2 port-based VLAN membership
  - 802.1q tag

  Resetting the priority has a local effect only. The priority controls the hardware forwarding queue the device uses to forward the packet but does not change the contents of the packet's 802.1p or IP ToS fields. For information about directly resetting the priority, see "Assigning QoS Priorities to Traffic" on page 2-11.

  This QoS method is available in releases earlier than 07.6.01 as well as in 07.6.01 and later.
- If the packet matches an ACL that explicitly sets the priority, the priority specified by the ACL is used. You can set a packet's priority using the following ACL options:
  - **priority** – Assigns traffic that matches the ACL to a hardware forwarding queue. In addition to changing the internal forwarding priority, if the outgoing interface is an 802.1q interface, this option maps the specified priority to its equivalent 802.1p (CoS) priority and marks the packet with the new 802.1p priority.
  - **dscp-marking** – Marks the DSCP value in the outgoing packet with the value you specify.

  If you use an ACL on an interface, ToS-based QoS assumes that the ACLs will perform QoS for all packets except the packets that match the **permit ip any any** ACL.

  NOTE: These options are new beginning in software release 07.6.01. See "QoS Options for IP ACLs (Rule-Based ACLs)" on page 6-44.

Configuring ToS-Based QoS

To configure ToS-based QoS, perform the following tasks:

- Globally enable basic ToS-based QoS. This is the only required task for basic QoS. The interface-level tasks are required only if you want to configure advanced QoS features.
- Optionally, enable advanced ToS-based QoS on an interface. Once you enable the feature on an individual interface, you can configure the trust level and marking for traffic that is forwarded on that interface.
  - Optionally, specify the trust level for packets received on the interface.
  - Optionally, enable marking of packets received on the interface.
- Optionally, change the QoS mappings. You can change the following mappings:
  - CoS –> DSCP
  - IP Precedence –> DSCP
  - DSCP –> DSCP
  - DSCP –> internal forwarding priority

  The mappings are globally configurable and apply to all interfaces.

Enabling Basic ToS-Based QoS

To enable basic ToS-based QoS, enter the following command at the global CONFIG level of the CLI:

```
BigIron(config)# port-priority
BigIron(config)# write memory
BigIron(config)# end
BigIron# reload
```

Syntax: [no] port-priority

This command enables the feature on all interfaces.

NOTE: You must save the configuration and reload the software to place the change into effect. Also, enabling port-priority changes the source MAC address of all ARP packets to a virtual MAC address.

Enabling Advanced ToS-Based QoS

To enable advanced ToS-based QoS on an interface, enter the following command at the configuration level for the interface:

```
BigIron(config-if-1/1)# qos-tos
```

Syntax: [no] qos-tos

NOTE: You must use this command if you want to configure the trust level or marking.

NOTE: When port priority is enabled, Foundry devices will use the higher of the 802.1p and DSCP priority for its internal system priority. To override this internal system priority, apply the **qos-tos mark cos** command on the interface where advanced ToS-based QoS was enabled. The true internal system priority will then be based on how the **qos-tos trust** command is configured on that interface. (See “Specifying the Trust Level” on page 4-7.)

If the port for incoming traffic is a tagged port, then the outgoing 802.1p priority will be the "merged" priority even though the type of marking configured on the interface is actually lower than the “merged” priority for that port. If the port for the incoming traffic is an untagged port and the port for outgoing traffic is an 802.1p port, then the marking configured on these ports will be used. This is useful in the case where the port priority (via priority command under interface CLI) is higher and the user wants to mark all IP packets with a lower priority. Hence, the non-IP packets will continue to use the higher 802.1p priority.

Specifying the Trust Level

The trust level specifies where you want the device to get the QoS value for a packet received on the interface.

To set the trust level for an interface to IP Precedence, enter the following command at the configuration level for the interface:

```
BigIron(config-if-1/1)# qos-tos trust ip-prec
```

Syntax: [no] qos-tos trust cos | ip-prec | dscp

The **cos | ip-prec | dscp** parameter specifies the trust level.

- **cos** – The device uses the 802.1p (CoS) priority value in the packet’s Ethernet frame header. Use this trust option when you plan to mark the packet’s DSCP value based on the incoming 802.1p value.

  NOTE: This trust level is not supported on 10 Gigabit Ethernet modules.

- **ip-prec** – The device uses the three most-significant bits in the packet’s ToS field and interprets them as an IP precedence value. Use this trust option when the incoming packet is from a device that does not support

DSCP and you need to mark the packet for QoS on DSCP devices.

- **dscp** – The device uses the six most-significant bits in the packet's ToS field and interprets them as a DSCP value. This is the default.

Enabling Marking

Marking changes the value of an outbound packet's 802.1p priority field, ToS field, or both to match the results of the QoS mappings performed by the Foundry device. When you enable marking on an interface, the marking applies to packets that enter the device through that interface.

To enable marking on an interface, enter a command such as the following at the configuration level for the interface:

```
BigIron(config-if-1/1)# qos-tos mark cos
```

This command enables marking of the 802.1p field in the Ethernet frame.

Syntax: [no] qos-tos mark cos | dscp

The **cos | dscp** parameter specifies the type of marking.

- **cos** – The device changes the outbound packet's 802.1p priority value to match the results of the device's QoS mapping from the specified trust level.
- **dscp** – The device changes the outbound packet's DSCP value to match the results of the device's QoS mapping from the specified trust level.

Changing the QoS Mappings

The Foundry device maps a packet's 802.1p, IP Precedence, or DSCP value into a DSCP value, and maps that DSCP value to an internal forwarding priority. The default mappings are listed in "Default QoS Mappings" on page 4-3. To change QoS mappings, use the commands described in the following sections.

NOTE: The mappings are globally configurable and apply to all interfaces.

NOTE: To place a mapping change into effect, you must enter the **ip rebind-acl all** command at the global CONFIG level of the CLI after making the mapping change. This applies to mappings that are configured using the **qos-tos map** command.

Changing the CoS –> DSCP Mappings

The CoS –> DSCP mappings are used if the trust level is CoS and DSCP marking is enabled.

To change the CoS –> DSCP mappings, enter commands such as the following at the global CONFIG level of the CLI:

```
BigIron(config)# qos-tos map cos-dscp 0 33 25 49 17 7 55 41
BigIron(config)# ip rebind-acl all
```

This command configures the mappings displayed in the COS-DSCP map portion of the QoS information display.

```
BigIron(config-if-1/1)# show qos-tos

...portions of table omitted for simplicity...

COS-DSCP map:

     COS: 0 1 2 3 4 5 6 7
  ------------------------------------------------------
     dscp: 0 33 25 49 17 7 55 41
```

Syntax: [no] qos-tos cos-dscp <dscp0> <dscp1> <dscp2> <dscp3> <dscp4> <dscp5> <dscp6> <dscp7>

The <dscp1> ... <dscp8> parameters specify the DSCP values you are mapping to the eight CoS values. You must enter DSCP values for all eight CoS values, in order from CoS value 0 – 7.

Changing the IP Precedence –> DSCP Mappings

The IP precedence –> DSCP mappings are used if the trust level is IP Precedence and DSCP marking is enabled.

To change the IP precedence –> DSCP mappings, enter commands such as the following at the global CONFIG level of the CLI:

```
BigIron(config)# qos-tos map ip-prec-dscp 0 32 24 48 16 8 56 40
BigIron(config)# ip rebind-acl all
```

This command configures the mappings displayed in the IP Precedence-DSCP map portion of the QoS information display.

```
BigIron(config-if-1/1)# show qos-tos
```

...portions of table omitted for simplicity...

```
IP Precedence-DSCP map:

  ip-prec:     0     1     2     3     4     5     6     7
  ---------------------------------------------------------
     dscp:     0    32    24    48    16     8    56    40
```

For information about the rest of this display, see “Displaying Configuration Information” .

Syntax: [no] qos-tos map ip-prec-dscp <dscp1> <dscp2> <dscp3> <dscp4> <dscp5> <dscp6> <dscp7> <dscp8>

The <dscp1> ... <dscp8> parameters specify the DSCP values you are mapping to the IP precedence values. You must enter DSCP values for all eight IP precedence values, in order from IP precedence value 0 – 7.

Changing the DSCP –> DSCP Mappings

To change a DSCP –> DSCP mapping, enter a command such as the following at the global CONFIG level of the CLI:

```
BigIron(config)# qos-tos map dscp-dscp 0 10
BigIron(config)# ip rebind-acl all
```

This command changes the mapping of DSCP value 0 from 0 to 10.

Syntax: [no] qos-tos map dscp-dscp <old-dscp-value> [<old-dscp-value>...]
to <new-dscp-value> [<new-dscp-value>...]

You can change up to eight DSCP values in the same commend. Make sure you enter the old values and their new values in the same order.

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:

```
BigIron(config)# qos-tos map dscp-priority 0 2 3 4 to 1
BigIron(config)# qos-tos map dscp-priority 8 to 5
BigIron(config)# qos-tos map dscp-priority 16 to 4
BigIron(config)# qos-tos map dscp-priority 24 to 2
BigIron(config)# qos-tos map dscp-priority 32 to 0
BigIron(config)# qos-tos map dscp-priority 40 to 7
BigIron(config)# qos-tos map dscp-priority 48 to 3
BigIron(config)# qos-tos map dscp-priority 56 to 6
BigIron(config)# ip rebind-acl all
```

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.

```
BigIron(config-if-1/1)# 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
```

For information about the rest of this display, see “Displaying Configuration Information” on page 4-11.

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 a command such as the following at the global CONFIG level of the CLI:

```
BigIron(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:

- qosp3
- qosp2
- qosp1
- qosp0

Displaying Configuration Information

To display configuration information, enter the following command at any level of the CLI:

```
BigIron(config)# show qos-tos
Interface QoS , Marking and Trust Level:

i/f    | QoS       | Mark      | Trust-Level
-------+-----------+-----------+---------------
1      | No        | No        | L2 CoS
2      | No        | No        | L2 CoS
3      | No        | No        | L2 CoS
4      | No        | No        | L2 CoS
5      | No        | No        | L2 CoS
6      | No        | No        | L2 CoS
7      | No        | No        | L2 CoS
8      | No        | No        | L2 CoS
9      | No        | No        | L2 CoS
10     | No        | No        | L2 CoS
... <lines omitted for brevity>

49     | No        | No        | L2 CoS
50     | No        | No        | L2 CoS
ve5    | No        | No        | L2 CoS
ve8    | No        | No        | L2 CoS
ve18   | No        | No        | L2 CoS
ve22   | No        | No        | L2 CoS

COS-DSCP map:

     COS: 0 1 2 3 4 5 6 7
  ------------------------------------------------------------
     dscp: 0 8 16 24 32 40 48 56
```

```
IP Precedence-DSCP map:

  ip-prec: 0 1 2 3 4 5 6 7
  ---------------------------------------------------------
     dscp: 0 8 16 24 32 40 48 56

DSCP-Priority map: (dscp = d1d2)

     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

DSCP-DSCP map: (dscp = d1d2)

     d2| 0 1 2 3 4 5 6 7 8 9
  d1   |
  -----+----------------------------------------
     0 | 0 1 2 3 4 5 6 7 8 9
     1 | 10 11 12 13 14 15 16 17 18 19
     2 | 20 21 22 23 24 25 26 27 28 29
     3 | 30 31 32 33 34 35 36 37 38 39
     4 | 40 41 42 43 44 45 46 47 48 49
     5 | 50 51 52 53 54 55 56 57 58 59
     6 | 60 61 62 63
```

Syntax: show qos-tos

This command shows the following information.

Table 4.6: ToS-Based QoS Configuration Information

This Field...	Displays...
Interface QoS, Marking and Trust Level information	
i/f	The interface
QoS	The state of ToS-based QoS on the interface. The state can be one of the following: • No – Disabled • Yes – Enabled
Mark	The marking type enabled on the interface. The marking type can be any of the following: • COS – CoS marking is enabled. • DSCP – DSCP marking is enabled. • No – Marking is not enabled.

Table 4.6: ToS-Based QoS Configuration Information (Continued)

This Field...	Displays...
Trust-Level	The trust level enabled on the interface. The trust level can be one of the following: • DSCP • IP Prec • L2 CoS
CoS-DSCP map	
COS	The CoS (802.1p) values.
dscp	The DSCP values to which the device maps the CoS values above.
IP Precedence-DSCP map	
ip-prec and dscp	The IP precedence -> DSCP mappings that are currently in effect.
DSCP-Priority map	
d1 and d2	The DSCP -> forwarding priority mappings that are currently in effect.
DSCP-DSCP map	
d1 and d2	The DSCP -> DSCP mappings that are currently in effect.

Chapter 5
Layer 2 ACLs

NOTE: This feature is available on the following devices:

- Devices with JetCore modules running Service Provide software release 09.1.00 or later
- BigIron MG8 and NetIron 40G running software release 02.1.00 and later
- NetIron IMR 640 running software release 02.0.02 and later
- Devices with JetCore modules running Enterprise software release 08.0.00 and later

Layer 2 Access Control Lists (ACLs) filter incoming traffic based on Layer 2 MAC header fields in the Ethernet/ IEEE 802.3 frame. Specifically, Layer 2 ACLs filter incoming traffic based on any of the following Layer 2 fields in the MAC header:

- Source MAC address and source MAC mask
- Destination MAC address and destination MAC mask
- VLAN ID
- Ethernet type

The Layer 2 ACL feature differs from the existing software-based MAC address filters. MAC address filters use the CPU to filter traffic; therefore, performance is limited by the CPU's processing power. Layer 2 ACLs are implemented in JetCore hardware and can thus filter traffic at line-rate speed.

Filtering Based on Ethertype

For JetCore devices:

JetCore Layer 2 ACLs can filter traffic based on protocol types of a frame. Depending on the type of traffic to filter, you can select a specific Ethertype (etype) on which to filter. There are different etypes for IP and IPX traffic, which provides flexibility to filter on packet details that are beyond Layer 2. For a list of the etypes supported with Layer 2 ACLs, see "Configuring Layer 2 ACLs" on page 5-3.

For each Layer 2 ACL etype entry bound to a port, a Content Addressable Memory (CAM) entry is written to the corresponding CAM. You can conserve CAM space by configuring only the Layer 2 ACLs needed. For instance, to filter only IPV4-Len-5 traffic, specify that particular etype. This results in one CAM entry. Configuration examples are provided in the section "Configuring Layer 2 ACLs" on page 5-3

For the BigIron MG8 and NetIron 40G:

In release 02.1.00 for the BigIron MG8 and NetIron 40G, you can configure Layer 2 ACLs to use the **etype** argument to filter on the following etypes:

- IPv4-15 (Etype=0x0800, IPv4, HeaderLen 20 bytes)
- ARP (Etype=0x0806, IP ARP)
- IPv6 (Etype=0x86dd, IP version 6)

For the NetIron IMR 640

In release 02.0.02 for the NetIron IMR 640, you can configure L2 ACLs to use the etype argument to filter on the following etypes (Ethertype):

- arp – The IP ARP Ethertype.
- ipv4 – The IP version 4 Ethertype with header length= 20 bytes.
- ipv6 – The IP version 6 Ethertype.

Syntax: [no] access-list <num> permit | deny any any any etype [arp | ipv4-L5 | ipv6]

The <num> parameter indicates the ACL number and must be from 400 to 499.

The **etype** parameter indicates that you are filtering on Layer 2 Ethertypes.

The **arp** parameter directs the ACL to permit or deny based upon the IP ARP Ethertype (Etype=0x0806).

The **ipv4-L5** parameter directs the ACL to permit or deny based upon the IP version 4 Ethertype (Etype=0x0800).

The **ipv6** parameter directs the ACL to permit or deny based upon the IP version 6 Ethertype (Etype=0x96DD).

Configuration Rules and Notes

For JetCore Devices

- You cannot bind Layer 2 ACLs and IP ACLs to the same port. However, you can configure one port on the device to use Layer 2 ACLs and another port on the same device to use IP ACLs.
- You cannot bind a Layer 2 ACL to a virtual interface.
- The Layer 2 ACL feature cannot perform SNAP and LLC encapsulation type comparisons. To implement these features, use MAC address filters. You can bind MAC filters and Layer 2 ACLs on the same port, however, the device will process the traffic in software instead of in hardware.
- When MAC address filters and Layer 2 ACLs are enabled on the same port, MAC address filter processing precedes Layer 2 ACL processing; the device either forwards or drops the traffic based on the MAC filter policies, and the traffic is not subject to Layer 2 ACL processing.
- By default, when Layer 2 ACLs are enabled on a port, the device filters traffic in hardware. However, when other CPU-based features, such as Net-flow and Adaptive-Rate-Limiting are also enabled on the port, traffic is sent to the CPU for additional processing and the Layer 2 ACLs are also processed in software. Note that the performance in this case is limited by the CPU cycles.
- By default, the device processes broadcast traffic in software. Filtering of broadcast packets is not handled by the hardware.
- You can use Layer 2 ACLs to block management access to the Foundry device. For example, you can use a Layer 2 ACL clause to block a certain host from establishing a connection to the device through Telnet.

For the BigIron MG8 and NetIron 40G

You can configure Layer 2 ACLs on BigIron MG8 and NetIron 40G software running software release release 02.1.00 and later

- You cannot bind Layer 2 ACLs and IP ACLs to the same port. However, you can configure one port on the

device to use Layer 2 ACLs and another port on the same device to use IP ACLs.

- You cannot bind a Layer 2 ACL to a virtual interface.
- By default, when Layer 2 ACLs are enabled on a port, the device filters traffic in hardware.

Configuring Layer 2 ACLs

Configuring a Layer 2 ACL is similar to configuring standard and extended ACLs. Layer 2 ACL table IDs range from 400 to 499, for a maximum of 100 configurable Layer 2 ACL tables. Within each Layer 2 ACL table, you can configure from 64 (default) to 256 clauses. Each clause or entry can define a set of Layer 2 parameters for filtering. Once you completely define a Layer 2 ACL table, you must bind it to the interface for filtering to take effect.

The Foundry device evaluates traffic coming into the port against each ACL clause. When a match occurs, the device takes the corresponding action. Once a match entry is found, the device either forwards or drops the traffic, depending upon the action specified for the clause. Once a match entry is found, the device does not evaluate the traffic against subsequent clauses.

By default, if the traffic does not match any of the clauses in the ACL table, the device drops the traffic. To override this behavior, specify a “permit any any…” clause at the end of the table to match and forward all traffic not matched by the previous clauses.

NOTE: Use precaution when placing entries within the ACL table. The Layer 2 ACL feature does not attempt to resolve conflicts and assumes you know what you are doing.

Configuration Considerations

When configuring Layer 2 ACLs, consider the following:

- Layer 2 ACLs are not supported on virtual routing interfaces.
- You cannot edit or modify an existing Layer 2 ACL clause. If you want to change the clause, you must delete it first, then re-enter the new clause.
- You cannot add remarks to a Layer 2 ACL clause.

Creating a Layer 2 ACL Table

You create a Layer 2 ACL table by defining a Layer 2 ACL clause.

To create a Layer 2 ACL table, enter commands (clauses) such as the following at the Global CONFIG level of the CLI. Note that you can add additional clauses to the ACL table at any time by entering the command with the same table ID and different MAC parameters.

Example for JetCore devices:

```
NetIron4000(config)# access-list 400 deny any any any etype appletalk
NetIron4000(config)# access-list 400 deny any any any etype ipx-raw
NetIron4000(config)# access-list 400 deny any any any etype ipx-snap
NetIron4000(config)# access-list 400 deny any any any etype ipx-1lc
NetIron4000(config)# access-list 400 permit any any 100
```

This configuration creates a Layer 2 ACL with an ID of 400. When applied to an interface, this Layer 2 ACL table will deny all AppleTalk and IPX traffic, and permit all other traffic in VLAN 100.

Example for the BigIron MG8 and NetIron 40G:

```
BigIron MG8(config)# access-list 400 deny any etype arp
BigIron MG8(config)# access-list 400 permit any any 100
```

This configuration creates a Layer 2 ACL with an ID of 400. When applied to an interface, this Layer 2 ACL table will deny all ARP traffic and permit all other traffic in VLAN 100.

For more examples of valid Layer 2 ACL clauses, see “Example Layer 2 ACL Clauses” on page 5-5.

Syntax: [no] access-list <num> permit | deny <src-mac> <mask> | any <dest-mac> <mask> | any [<vlan-id> | any [etype <etype-str>] [log-enable]]

The <num> parameter specifies the Layer 2 ACL table that the clause belongs to. The table ID can range from 400 to 499. You can define a total of 100 Layer 2 ACL tables.

The **permit** | **deny** argument determines the action to be taken when a match occurs.

The <src-mac> <mask> | **any** parameter specifies the source MAC address. You can enter a specific address and a comparison mask or the keyword **any** to filter on all MAC addresses. Specify the mask using F’s 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 clause matches all source MAC addresses that contain “aabb” as the first two bytes and any values in the remaining bytes of the MAC address. If you specify **any**, you don’t need to specify a mask and the clause 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.

The optional <vlan-id> | **any** parameter specifies the vlan-id to be matched against the vlan-id of the incoming packet. You can specify **any** to ignore the vlan-id match.

The optional **etype** <etype-str> argument specifies the Ethernet type field of the incoming packet in order for a match to occur.

The <etype-str> for the JetCore devices can be one of the following keywords:

- IPV4 (Etype=0x0800, IP version 4)
- IPV4-Len-5 (Etype=0x0800, IPV4, HeaderLen 20 bytes)
- IPV4-IGMP (Etype=0x0800, IPV4, Protocol=2)
- IPV4-IGMP-Len-5 (Etype=0x0800, IPV4-L5, Protocol=2)
- ARP (Etype=0x0806, IP ARP)
- IPX-Raw (Etype<1536, DSAP-SSAP = 0xFFFF)
- IPX-LLC (Etype<1536, DSAP-SSAP = 0xE0E0)
- IPX-SNAP (Etype<1536, DSAP-SSAP = 0xAAAA_03, Snap_Etype=0x8137)
- IPX-8137 (Etype=0x8137)
- AppleTalk (Etype<1536, DSAP-SSAP = 0xAAAA_03, Snap_Etype=0x809B)
- Apple Talk ARP (Etype<1536, DSAP-SSAP = 0xAAAA_03, Snap_Etype=0x80F3)
- Net Bios (Etype<1536, DSAP-SSAP = 0xF0F0/0xF0F1)
- IP SNAP (Etype<1536, DSAP-SSAP = 0xAAAA_03, Snap_Etype=0x0800)
- IPV6 (Etype=0x86DD, IP version 6)

The <etype-str> for the BigIron and NetIron 40G running software release 02.1.00 and later can be one of the following keywords:

- IPv4-15 (Etype=0x0800, IPv4, HeaderLen 20 bytes)
- ARP (Etype=0x0806, IP ARP)
- IPv6 (Etype=0x86dd, IP version 6)

The optional <log-enable> parameter enables the logging mechanism. The device accepts this command only when a **deny** clause is configured. When you enable logging for a Layer 2 ACL, all traffic matching the clause is sent to the CPU for processing and traffic is denied by the CPU. The CPU creates a log entry for the first packet that is denied and once every 10 seconds thereafter. The logging mechanism includes sending SNMP traps and log messages to the Syslog servers and writing the log entry to the log buffer on the device.

In addition, for the BigIron MG8 and NetIron 40G, if specified with a 'permit' action, the log-enable keyword is ignored and the user is warned that he cannot log permit traffic.

NOTE: Traffic denied by the implicit deny mechanism is not subject to logging. The implicit deny mechanism kicks in when the traffic does not match any of the clauses specified and there is no **permit any any** clause specified at the end.

Use the [no] parameter to delete the Layer 2 ACL clause from the table. When all clauses are deleted from a table, the table is automatically deleted from the system.

Example Layer 2 ACL Clauses

The following shows some examples of valid Layer 2 ACL clauses for JetCore devices:

```
NetIron4000(config)# access-list 400 permit any any
NetIron4000(config)# access-list 400 permit any any log-enable
NetIron4000(config)# access-list 400 permit any any 100
NetIron4000(config)# access-list 400 permit any any 100 log-enable
NetIron4000(config)# access-list 400 permit any any any
NetIron4000(config)# access-list 400 permit any any any log-enable
NetIron4000(config)# access-list 400 permit any any 100 etype ipv4
NetIron4000(config)# access-list 400 permit any any 100 etype ipv4 log-enable
```

The following shows an example of a valid Layer 2 ACL clause for the BigIron and NetIron 40G:

```
BigIron MG8(config)# access-list 400 permit any any 100 etype ipv4
```

Inserting and Deleting Layer 2 ACL Clauses

You can make changes to the Layer 2 ACL table definitions without unbinding and rebinding the table from an interface. For example, you can add a new clause to the ACL table, delete a clause from the table, delete the ACL table, etc.

Binding a Layer 2 ACL Table to an Interface

To enable Layer 2 ACL filtering, bind the Layer 2 ACL table to an interface. Enter a command such as the following at the Interface level of the CLI:

```
NetIron4000(config)# int e 4/12
NetIron4000(config-int-e100-4/12)# mac access-group 400 in
```

Syntax: [no] mac access-group <num> in

The <num> parameter specifies the Layer 2 ACL table ID to bind to the interface. Enter 400-499.

Increasing the Maximum Number of Clauses per Layer 2 ACL Table

You can increase the maximum number of clauses configurable within a Layer 2 ACL table. You can specify a maximum of 256 clauses per table. The default value is 64 clauses per table.

To increase the maximum number of clauses per Layer 2 ACL table, enter a command such as the following at the Global CONFIG level of the CLI:

```
NetIron4000(config)# system-max l2-acl-table-entries 200
```

Syntax: system-max l2-acl-table-entries <max>

The <max> parameter specifies the maximum number of clauses per Layer 2 ACL. Enter a value from 64 to 256.

Viewing Layer 2 ACLs

Use the **show access-list** command to monitor configuration and statistics and to diagnose Layer 2 ACL tables. The following shows an example output.

```
NetIron 4000(config)# show access-list 400

L2 MAC Access List 400:
  permit any any 100 etype ipv4
  deny any any any etype appletalk
  deny any any any etype ipx-raw
  deny any any any etype ipx-snap
  deny any any any etype ipx-llc


BigIron MG8(config)# show access-list 400

L2 MAC Access List 400:
  permit any any 100 etype ipv4
  deny any any any etype arp
```

Syntax: show access-list <num>

The <num> parameter specifies the Layer 2 ACL table ID.

Example of Layer 2 ACL Deny by MAC Address (Release 02.1.00 for the BigIron MG8 and NetIron 40G)

In the following example, an ACL is created that denies all traffic from the host with the MAC address 0012.3456.7890 being sent to the host with the MAC address 0011.2233.4455.

```
BigIron MG8(config)# access-list 401 deny 0012.3456.7890 ffff.ffff.ffff
0011.2233.4455 ffff.ffff.ffff
BigIron MG8(config)# access-list 401 permit any any
```

Using the mask, you can make the access list apply to a range of addresses. For instance if you changed the mask in the previous example from 0012.3456.7890 to ffff.ffff.fff0, all hosts with addresses from 0012.3456.7890 to 0012.3456.789f would be blocked. This configuration for this example is shown in the following:

```
BigIron MG8(config)# access-list 401 deny 0012.3456.7890 ffff.ffff.fffe
0011.2233.4455 ffff.ffff.ffff
BigIron MG8(config)# access-list 401 permit any any
```

Chapter 6
Access Control List

This chapter discusses Foundry's IP Access Control List (ACL) feature, which enables you to filter traffic based on the information in the IP packet header. Depending on the Foundry device, the device may also support Layer 2 ACLs, which filter traffic based on Lay 2 MAC header fields. For details on Layer 2 ACLs, see "Layer 2 ACLs" on page 5-1.

You can use IP ACLs to provide input to other features such as route maps, distribution lists, rate limiting, and BGP. 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. Also, if you use an ACL in a route map and you use a wildcard character as the source IP address, make sure you apply the route map to interfaces, not globally. Otherwise, a loop can occur. See the chapters for a specific feature for information on using ACLs as input to those features.

This chapter presents the following sections:

How Foundry Devices Process ACLs

There are two ways that ACLs are processed in Foundry devices: in software and in hardware.

Flow-based ACLs

Some Foundry process traffic that ACLs filter in software or CPU are called flow-based ACLs in this document. This type of ACL is also referred to as flow-based or CPU-based ACLs. Flow-based ACLs are useful when a very large number of unique ACLs are defined and these ACLs exceed the Content Addressable Memory (CAM) memory of a given module, but a subset of the defined ACLs will be used at any given time on a module. Table 6.1 lists the products and software releases that support flow-based ACLs.

Table 6.1: Flow-based ACL Support

Product	Software Releases
FastIron 4802	Software releases prior to 07.6.01 support flow-based ACLs only. Starting with release 07.6.01, you can configure an interface to use flow-based or rule-based ACL mode.
FastIron Edge Switch (FES)	01.0.00 and later
Devices that have IronCore modules	All
Devices that have JetCore modules	Software releases prior to 07.6.01 support flow-based ACLs only. Starting with release 07.6.01, you can configure an interface to use flow-based or rule-based ACL mode.

Flow-based ACLs work as follows:

When the device receives an IP packet, the device checks the receiving port's ACL CAM entries for an entry with the same address information as the packet.

- If the CAM contains a matching entry, the device takes the action specified by the entry (permit or deny).

NOTE: In flow-based ACLs, CAM entries are not programmed when you apply an ACL to an interface. CAM entries are created by the CPU when a a permit claus in the ACL bound the interface, as described below. The Layer 4 CAM entries programmed by the CPU for ACL matches age out if unused for 70 seconds.

NOTE: The CAM can contain entries for ACLs with deny actions only if you enable this support by entering the **hw-drop-acl-denied-packet** command.

- If the CAM does not contain a matching entry, the device sends the packet to the CPU for ACL comparison.
  - If the packet matches an ACL applied to inbound traffic on the port and the ACL has the permit action, the CPU programs an ACL permit entry into the Layer 4 CAM for the port that received the packet. The CAM entry contains the packet's address information. All subsequent packets from this flow will match the CAM and are forwarded in hardware.
  - If the packet matches an ACL applied to inbound traffic on the port and the ACL has the deny action, the CPU drops the packet but does not program an entry into the Layer 4 CAM, unless you have enabled the CPU to do so by entering the **hw-drop-acl-denied-packet** command.
  - If the packet does not match any of the inbound ACLs on the interface (and therefore matches an implicit **deny ip any any**), the CPU drops the packet. The CPU does not program an entry into the Layer 4 CAM, unless you have enabled the CPU to do so by entering the **hw-drop-acl-denied-packet** command.
- If the packet's outbound interface has an ACL applied to the outbound traffic direction, the device sends the packet to the CPU for filtering and either drops the packet or forwards the packet on the outbound interface, depending on the results of the ACL comparison.

Configuration Guidelines for Flow-Based ACLs

Refer to the following guidelines when configuring flow-based ACLs.

- For optimal performance, apply deny ACLs to inbound ports instead of outbound ports. This way, traffic is dropped as it tries to enter the Foundry device, instead of being dropped after it has been forwarded internally to the outbound port.
- Outbound ACLs do not filter broadcast traffic or any traffic (including ICMP replies) generated by the Foundry device itself.
- You cannot use ACLs to filter based on MAC information or Quality of Service (QoS) information.

  To filter based on MAC information, see the "Defining MAC Address Filters" section in the "Configuring Basic Features" chapter of the *Foundry Switch and Router Installation and Basic Configuration Guide*.

  To filter based on QoS information, see "Assigning IP and Layer 4 Sessions to Priority Queues" on page 2-18.
- On Layer 3 Switches, ACLs do not apply to traffic that is switched between one port and another in the same VLAN.
- ACLs on broadcast, multicast, and unknown unicast traffic on outbound ACL is not supported.
- On the FastIron Backbone switch, management access ACLs are supported, but packet forwarding ACLs are not supported.

Rule-Based ACLs

Some Foundry devices process the traffic that ACLs filter in hardware. This document refers to this type of ACLs as rule-based ACLs. This type of ACL is also called rule-based ACL. These ACLs are programmed into hardware at startup or as a new ACL is entered. Table 6.2 lists the products and software releases that support rule-based ACLs.

Table 6.2: Rule-Based ACL Support

Product	Software Releases
Devices with 10-Gigabit Ethernet modules	07.6.01 and later[a]
FastIron 4802	07.6.01 and later[a]

Table 6.2: Rule-Based ACL Support

Product	Software Releases
FastIron Edge Switch (FES) X-Series	01.0.00 and later[b]
Devices with JetCore modules	07.6.01 and later[a]
NetIron 40G[c]	01.0.00 and later
BigIron MG8[c]	01.0.00 and later
NetIron IMR 640	02.0.02 and later

a. Starting with release 07.6.01, you can configure an interface to use flow-based or rule-based ACL mode.

b. The FES X-Series supports rule-based ACLs only. It does not support flow-based ACLs.

c. Beginning with BigIron MG8 software release 02.0.02 and NetIron 40G software release 02.0.01, IPv4, IPv6, and Outbound IPv4 ACLs are supported on certain interface modules. See the "Support for Software Features by TCAM Version Installed" section in the "Product Overview" chapter of the *Foundry BigIron Mg8 Switch Installation and Basic Configuration Guide*.

Rule-based ACLs program the ACL entries you assign to an interface into Content Addressable Memory (CAM) space allocated for the port(s). 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.

In releases prior to 07.6.01, Foundry devices use the flow-based mode to process ACLs. The first packets received for a given traffic flow (pair of source and destination addresses) are sent to the CPU. The device then makes an entry in the Layer 4 session table and uses the entry to permit or deny traffic for the flow.

In software releases 07.6.01 and later, you can configure a Foundry device to use the rule-based ACL mode or the flow-based mode. You can change the ACL mode on an individual interface basis. The rule-based mode is enabled by default on all interfaces.

NOTE: The FES X-Series supports rule-based ACLs only. It does not support flow-based ACLs.

Rule-based ACLs are supported on physical interfaces, trunk groups, and virtual routing interfaces.

Configuration Guidelines for Rule-Based ACLs: General Guidelines

- Rule-based ACLs are supported on all JetCore Ethernet ports, on NPA POS OC-48 ports, on 10 Gigabit Ethernet ports and on BigIron MG8 and on NetIron 40G.
- Beginning with Terathon Ironware release 02.1.00, you can have up to 40K (40,960) ACL statements on a NetIron 40G and 4K (4096) ACL statements on a BigIron MG8.
- Rule-based ACLs are supported on physical interfaces, trunk groups, and virtual routing interfaces.
- Rule-based ACLs are supported only for inbound traffic except for the NetIron IMR 640 and if the outbound interface is an NPA POS OC-48 interface.
- If the outbound interface is an NPA POS OC-48 interface, the device does not change the ACL mode. If you apply an ACL to outbound traffic on another port type (other than NPA OC-48c), the device changes to flow-based ACLs.
- Rule-based ACLs support only one ACL per port. The ACL of course can contain multiple entries (rules). For example, rule-based ACLs do not support ACLs 101 and 102 on port 1, but rule-based ACLs do support ACL 101 containing multiple entries.
- If you change the content of an ACL (add, change, or delete entries), you must remove and then reapply the ACL to all the ports that use it. Otherwise, the older version of the ACL remains in the CAM and continues to

be used. You can easily re-apply ACLs using the **ip rebind-acl** <num> | <name> | **all** command. See "Applying an ACLs to Interfaces" on page 6-39.

NOTE: Foundry recommends that you also remove and reapply a changed ACL even when you are using the flow-based mode.

- If you or the software changes the ACL mode (between rule-based and flow-based), you must reapply the ACLs to the ports. Use the **ip rebind-acl** <num> | <name> | **all** command.
- ACL statistics are not supported with rule-based rate limiting. This feature relies on ACL information provided by the CPU, and thus requires flow-based ACLs. If you enable ACL statistics (by entering the **enable-acl-counter** command), the device automatically changes the ACL mode on all ports to flow-based ACLs.
- If you use the <icmp-type> parameter with an extended ACL, the device uses the CPU to filter packets using the ACL. The CPU is required to filter the ICMP message type.
- For a tagged port that is a member of multiple virtual routing interfaces, you must use the same ACL on all the port's virtual routing interfaces. Alternatively, if you need to use different ACLs, you can use flow-based ACLs instead on all the port's virtual routing interfaces.
- The software automatically disables rule-based ACLs and enables flow-based ACLs on an interface if one of the following occurs:
  - If you apply an ACL to the outbound traffic direction on the interface.

  NOTE: The ACL mode is not changed if the outbound interface is an NPA POS OC-48 port.

  - If there is not enough CAM space to hold all the ACL entries in the ACL applied to the port. All ACL entries for a port must fit in the CAM space allocated by the device for the port's ACLs. If all the ACL's entries do not fit into the port's CAM, the device disables rule-based ACLs on the port and enables flow-based ACLs instead. The device also generates a Syslog message to inform you of the change. See "ACL Logging for Rule-Based ACLs" on page 6-41.

    If this occurs, remove the ACL from the port, then either reduce the number of entries in the ACL to fit into the CAM space or adjust the CAM allocations on the device's ports, to hold the ACL.
  - If you enable any of the following features on the interface. To change back to the rule-based ACL mode in this case, you must disable the feature that caused the port to change to the flow-based ACL mode.
    – Network Address Translation (NAT)
    – Protection against ICMP or TCP Denial-of-Service (DoS) Attacks
    – ACL Logging
- You can use PBR and rule-based ACLs on the same port. However, Foundry recommends that you use exactly the same ACL for each feature. Otherwise, it is possible for the ACL's Layer 4 CAM entry to be programmed incorrectly and give unexpected results.

Configuration Guidelines for Rule-Based ACLs: FES-X Series

The following ACL features and options are not supported on the FES X-Series:

- Applying an ACL to a Subset of Ports on a Virtual Interface
- Enabling CPU filtering of all fragmented packets on a port (**ip access-group frag inspect** command)
- Configuring a port to drop all packet fragments (**ip access-group frag deny** command)
- ACL logging
- Flow-based ACLs
- ACL statistics
- ACL-based rate limiting

Configuration Guidelines for Rule-Based ACLs: NetIron IMR 640

- In release 02.0.02 for the NetIron IMR 640, ACLs are supported on the following MPLS VPN Endpoints:
  - IPv4 and IPv6 inbound ACLs are not supported on VPLS and VLL endpoints.
  - IPv4 ACL-based rate limiting is not supported on VPLS and VLL endpoints
  - Layer-2 ACLs and Layer-2 ACL-based rate limiting is not supported on Layer-3 VPNs
  - TOS/DSCP marking using inbound ACLs is not supported on Layer-3 VPNs.
  - PBR policies are not supported on Layer-3 VPNs.
- Multi-Service IronWare release 02.0.02 provides support for up to 40K (40,960) ACL statements on the NetIron IMR 640.
- Release 02.0.02 for the NetIron IMR 640 supports both inbound and outbound IPv4 traffic.

How Fragmented Packets are Processed

The descriptions for flow-based and rule-based ACLs above apply to non-fragmented packets. In 07.6.01 and later, the default processing of fragments by both flow-based and rule-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, one of the following occurs:
  - If the device has a CAM entry for the packet (or for previous packets in the same flow), and has not been configured to send the fragments to the CPU, the device uses the CAM entry to forward the fragments in hardware.

    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. However, for stricter fragment control, you can send fragments to the CPU for filtering.
  - If the device is configured to send fragments to the CPU for filtering, the device compares the source and destination IP addresses to the ACL entries that contain Layer 4 information.

    —If the fragment's source and destination addresses exactly match an ACL entry that has Layer 4 information, the device assumes that the ACL entry is applicable to the fragment and permits or denies the fragment according to the ACL entry. The device does not compare the fragment to ACL entries that do not contain Layer 4 information.

    —If both the fragment's source and destination addresses do not exactly match an ACL entry, the device skips the ACL entry and compares the packet to the next ACL entry. This is true even if either the source or destination address (but not both) does exactly match an ACL entry.

    —If the source and destination addresses do not exactly match any ACL entry on the applicable interface, the device drops the fragment.

NOTE: By default, 10 Gigabit Ethernet modules also forward the first fragment instead of using the ACLs to permit or deny the fragment.

You can modify the handling of denied fragments. In addition, you can throttle the fragment rate on an interface that used rule-based ACLs. See "Dropping All Fragments That Exactly Match a Flow-Based ACL" on page 6-48 and "Enabling ACL Filtering of Fragmented Packets" on page 6-51.

Disabling or Re-Enabling Access Control Lists (ACLs)

Commands used to enable or disable ACLs are different for flow-based ACLs and for rule-based ACLs.

For Flow-Based ACLs

A Layer 3 Switch that supports flow-based ACLs cannot actively use both IP access policies and ACLs for filtering IP traffic. When you boot a Layer 3 Switch with software release 06.5.00 or higher, the software checks the device's startup-config file for **ip access-policy-group** commands, which associate IP access policies with ports. If the software finds an **ip access-policy-group** command in the file, the software disables all packet-forwarding ACLs (those associated with specific ports) and also prevents you from applying an ACL to a port.

The next time you save the startup-config file, the software adds the following command near the top of the file, underneath the **ver** (software version) statement:

```
ip dont-use-acl
```

This command disables all packet-forwarding ACLs that are associated with specific ports and also prevents you from associating an ACL with a port. However, the command does not remove existing ACLs from the startup-config file. In addition, the command does not affect ACLs used for controlling management access to the device.

If the device supports flow-based ACLs and you want to use ACLs instead of IP acess policies, you must enable ACL mode using the **no ip dont-use-acl** command.

Enabling ACL Mode

If you try to apply an ACL to a port when the ACL mode is disabled (when the **ip dont-use-acl** command is in effect), a message is displayed, as shown in the following CLI example:

```
BigIron(config-if-e1000-1/1)# ip access-group 1 out
Must enable ACL mode first by using no ip dont-use-acl command and removing all ip
access-policy-group commands from interfaces, write memory and reload
```

As the message states, if you want to use ACLs, you must first enable the ACL mode. To do so, use either of the following methods.

USING THE CLI

To enable the ACL mode, enter the following commands:

```
BigIron(config-if-e1000-1/1)# exit
BigIron(config)# no ip dont-use-acl
BigIron(config)# write memory
BigIron(config)# end
BigIron# reload
```

Syntax: no ip dont-use-acl

The **write memory** command removes the **ip dont-use-acl** command from the startup-config file. The **reload** command reloads the software. When the software finishes loading, you can apply ACLs to ports.

The commands that configure the IP access policies and apply them to ports remain in the startup-config file in case you want to use them again, but they are disabled. If you later decide you want to use the IP access policies again instead of ACLs, you must disable the ACL mode again. See the following section.

USING THE WEB MANAGEMENT INTERFACE

NOTE: The ACL feature is automatically enabled on a Layer 2 Switch.

1. Log on to the Layer 3 Switch using a valid user name and password for read-write access. The System configuration panel is displayed.
2. Click on the plus sign next to Configure in the tree view to expand the list of configuration options.
3. Click on the plus sign next to IP in the tree view to expand the list of IP option links.
4. Click on the General link to display the IP configuration panel.

5. Select the Enable radio button next to Access Control List.
6. Click the Apply button to save the change to the device's running-config file.
7. 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.

Disabling ACL Mode

If the ACL mode is enabled, a message is displayed when you try to apply an IP access policy to a port, as shown in the following CLI example:

```
BigIron(config-if-e1000-1/1)# ip access-policy-group 1 in
Must disable ACL mode first by using ip dont-use-acl command, write memory and
reload
```

To use the IP access policies, you first must disable the ACL mode using either of the following methods.

USING THE CLI

To disable the ACL mode, enter the following commands:

```
BigIron(config-if-e1000-1/1)# exit
BigIron(config)# ip dont-use-acl
BigIron(config)# write memory
BigIron(config)# end
BigIron# reload
```

Syntax: ip dont-use-acl

The **write memory** command removes the **ip dont-use-acl** command from the startup-config file. The **reload** command reloads the software. When the software finishes loading, you can apply ACLs to ports.

The commands that configure the IP access policies and apply them to ports remain in the startup-config file in case you want to use them again, but they are disabled. If you later decide you want to use the IP access policies again instead of ACLs, you must disable the ACL mode again. See the following section.

USING THE WEB MANAGEMENT INTERFACE

NOTE: The ACL feature cannot be disabled on a Layer 2 Switch.

1. Log on to the Layer 3 Switch using a valid user name and password for read-write access. The System configuration panel is displayed.
2. Click on the plus sign next to Configure in the tree view to expand the list of configuration options.
3. Click on the plus sign next to IP in the tree view to expand the list of IP option links.
4. Click on the General link to display the IP configuration panel.
5. Select the Disable radio button next to Access Control List.
6. Click the Apply button to save the change to the device's running-config file.
7. 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.

Disabling or Re-Enabling Rule-Based ACLs

By default, the devices that support rule-based ACLs enable these ACLs on all ports. The device also can disable rule-based ACLs based on the conditions described in "Configuration Guidelines for Rule-Based ACLs: General Guidelines" on page 6-4. You also can manually change the ACL mode on an interface.

Disabling rule-based ACLs on some of the ports is useful if some ports have large ACLs (ACLs with many entries) while other ports have few ACL entries. By disabling rule-based ACLs on the ports that have few or no ACL entries, you can ensure that the ports that do have ACL entries will have enough CAM space for the ACL entries.

You also might want to disable rule-based ACLs if the ACL entries on a port are used infrequently. In this case, you can conserve CAM entries for other features or other ports with minimal performance impact, since the ACL activity is low.

NOTE: You can determine the ports that have high ACL usage by disabling rule-based ACLs on all the ports, allowing the device to operate using flow-based ACLs, then displaying ACL accounting information. To enable ACL accounting, enter the **enable-acl-counter** command at the global CONFIG level. To display the ACL accounting information, enter the **show access-list all** command.

To disable rule-based ACLs on an interface, enter the following command at the configuration level for the port:

```
BigIron(config-if-1/1)# ip access-group flow-mode
```

Syntax: [no] ip access-group flow-mode

To re-enable rule-based ACLs on the port, enter the following command:

```
BigIron(config-if-1/1)# no ip access-group flow-mode
```

Default ACL Action

The default action when no ACLs is 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.

NOTE: Do not apply an empty ACL (an ACL ID without any corresponding entries) to an interface. If you accidentally do this, the software applies the default ACL action, deny all, to the interface and thus denies all traffic.

Types of IP ACLs

Flow-based or rule-based ACLs can be configured as standard or extended ACLs. A standard ACL permits or denies packets based on source IP address. An extended ACL permits or denies packets based on source and destination IP address and also based on IP protocol information.

Standard or extended ACLs can be numbered or named. Standard numbered ACLs have an idea of 1 – 99. Extended numbered ACLs are numbered 100 – 199. IDs for standard or extended ACLs can be a character string. In this document, ACLs with a string ID is called a named ACL.

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.

  NOTE: This is different from IP access policies. If you use IP access policies, you apply the individual policies to interfaces.

- ACL entry – An ACL entry are the filter commands associated with an ACL ID. These are also called "statements". The maximum number of ACL entries 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.
  - Up to 1024 entries are supported on Layer 3 Switches using Management 1, Management 2, or Management 3 modules.
  - Management 4 modules can support up to 4096 ACL entries.
  - Layer 3 switch code on devices with JetCore and 10-Gigabit Ethernet modules running software release 07.8.00 or later can support up to 8192 ACL entries.
  - FES devices support up to 4000 ACL entries.
  - 10-Gigabit ports on the FES X-Series support up to 1024 ACL statements. One-Gigabit ports on the FES-X Series support up to 1016 ACL entries.
  - This feature is not supported on the BigIron MG8 and NetIron 40G

You configure ACLs on a global basis, then apply them to the incoming or outgoing traffic on specific ports. You can apply only one ACL to a port's inbound traffic and only one ACL to a port's outbound traffic. 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.

Enabling Support for Additional ACL Statements

You can enable support for additional ACL statements on some devices.

Support for up to 4096 ACL Entries on Terathon Devices

Terathon release 02.1.00 and later allows you to enable additional ACL statements on the BigIron MG8 and NetIron 40G. Multi-Service IronWare release 02.0.02 provides support for up to 40K ACL statements on the NetIron IMR 640.

On the BigIron MG8

To enable the BigIron MG8 running release 02.1.00 and later to support 4,096 ACL statements, enter the following command at the Global CONFIG level of the CLI:

```
BigIron MG8(config)# system-max ip-filter-sys 4096
```

Syntax: [no] system-max ip-filter-sys <num>

<num> is a value from 64 to 4096. The default is 4K (4096).

On the NetIron 40G and NetIron IMR 640

Releases 02.1.00 and later for the NetIron 40G and releases 02.0.02 and later for the NetIron IMR 640 provide support for up to 40K (40,960) ACL statements.

To enable the NetIron 40G to support 40,960 ACL entries, enter the following command at the Global CONFIG level of the CLI:

```
NetIron 40G(config)# system-max ip-filter-sys 40960
```

Syntax: [no] system-max ip-filter-sys <num>

<num> is a value from 64 to 40960. The default is 4K (4096).

Support for up to 4096 ACL Entries on Other Foundry Devices

On other Foundry devices, you can configure up to 4096 ACL entries on devices that have enough space to hold a startup-config file that contains the ACLs.

NOTE: Support for 4096 ACL entries applies only to the NetIron Internet Backbone router, BigIron Layer 3 Switches with Management 4 modules, and FES devices. The PCMCIA flash card on the Management 4 module is required to store and load startup-config files containing the large number of ACLs.

You do not need to configure the device's memory for the increased support.

The feature is supported on all chassis Layer 3 Switches. However, the actual number of ACLs you can configure and store in the startup-config file depends on the amount of memory available on the device for storing the startup-config. To store 4096 ACLs in the startup-config file requires at least 250K bytes, which is larger than the space available on a device's flash memory module. To store this many ACLs, you need a Management 4 module with a PCMCIA flash card or a TFTP server.

You can load ACLs dynamically by saving them in an external configuration file on flash card or TFTP server, then loading them using one of the following commands:

- **copy slot1 | slot2 running** <from-name>
- **ncopy slot1 | slot2** <from-name> **running**
- **copy tftp running-config** <ip-addr> <filename>
- **ncopy tftp** <ip-addr> <from-name> **running-config**

In this case, the ACLs are added to the existing configuration. See the "Dynamic Configuration Loading" section in the "Updating Software Images and Configuration Files" chapter of the *Foundry Switch and Router Installation and Basic Configuration Guide*.

Support for up to 8192 ACL Entries

Software releases 07.8.00 and later can support up to 8192 ACL statements on devices with JetCore and 10 Gigabit Ethernet modules running Layer 3 switch code.

- This feature is supported starting in software release 07.8.00.
- This feature is supported on devices with JetCore and 10-Gigabit Ethernet modules, running Layer 3 switch code.
- For rule-based ACLs, before configuring this feature, you must check the Layer 4 CAM space to ensure there is enough space. To check the Layer 4 CAM space, enter the command **show cam-partition**. To increase the Layer 4 CAM partition, use the **cam-partition** command. For more information about CAM partitioning, see the *Foundry Diagnostic Guide*.
- FastIron devices do not have enough CAM space to support 8000 rule-based ACL entries.

To enable the Foundry device to support up to 8192 ACL entries, enter the following command at the Global CONFIG level of the CLI:

```
BigIron(config)# system-max ip-filter-sys 8192
```

Syntax: `[no] system-max ip-filter-sys <num>`

Enter a value from 64 to 8192 for <num>.

ACL Entries and the Layer 4 CAM

Flow-based ACLs and rule-based ACLs both use Layer 4 CAM entries.

Aging Out of Entries in the Layer 4 CAM

On most Foundry devices, the device permanently programs rule-based ACLs into the CAM. The entries never age out.

However, on devices with IronCore and JetCore modules running software release 07.5.04 and later, the device does age out Layer 4 CAM entries for flow-based ACLs. A Layer 4 CAM entry for a flow-based ACL ages out if the

entry is unused for 70 seconds. The age time is not configurable. After an entry ages out, its CAM space becomes available for other ACL entries or other features that use the Layer 4 CAM.

Displaying the Number of Layer 4 CAM Entries

To display the number of Layer 4 CAM entries used by each ACL, enter the following command:

```
BigIron(config)# 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.

Specifying the Maximum Number of CAM Entries for ACLs (Rule-Based ACLs)

For rule-based ACLs, you can adjust the allocation of Layer 4 CAM space for use by ACLs, on an IPC or IGC basis and on 10 Gigabit Ethernet modules. The new allocation applies to all the ports managed by the IPC or IGC or 10 Gigabit Ethernet module.

Most ACLs require one CAM entry for each ACL entry (rule). The exception is an ACL entry that matches on more than one TCP or UDP application port. In this case, the ACL entry requires a separate Layer 4 CAM entry for each application port on which the ACL entry matches.

Make sure you specify a maximum that is equal to or greater than the largest number of entries required by an ACL applied to any of the ports managed by the same IPC or IGC. For example, if port 1 on a FastIron 4802 will have an ACL that requires 250 entries, make sure 250 is the lowest number of entries you specify for any port on IPC 1 (the IPC that manages ports 1 – 24).

Rule-based ACLs use CAM partitions 1 and 2. The default number of entries that are allocated in each pool differs depending on the device. For more information about CAM partitions, see the "Changing CAM Partitions" chapter in the *Foundry Diagnostic Guide*.

To specify the maximum number of CAM entries the device can allocate for rule-based ACLs, enter commands such as the following:

```
BigIron(config)# interface ethernet 1/1
BigIron(config-if-1/1)# ip access-group max-l4-cam 50
```

This command allows up to 50 ACL entries on each port managed by the IPC or IGC that manages port 1/1.

Syntax: [no] ip access-group max-l4-cam <num>

The <num> parameter specifies the number of CAM entries and can be from 10 – 2048. The default depends on the device.

The command is valid at the interface configuration level. However, the device applies the change to all ports managed by the same IPC or IGC. Regardless of the port number, when you save the change to the startup-config file, the CLI applies the command to the first port managed by the IPC or IGC. For example, if you enter the command on port 3 of a FastIron 4802, when you save the configuration change, the CLI enters the **ip access-group max-l4-cam** command under port 1 in the startup-config file.

NOTE: If you enter the command on more than one port managed by the same IPC or IGC, the CLI uses the value entered with the most-recent command for all the ports on the ICP or IGC.

ACL CAM Sharing for Inbound ACLs

NOTE: This feature applies to release 02.1.00 for the NetIron IMR 640.

In the previous release, inbound ACLs reserved CAM space for each instance of an ACL on each port that it is applied to. For example, if ACL 101 is bound to Gigabit Ethernet port 1/1 and port 1/5 which are on the same PPCR, a separate CAM space is allocated for each port. This is still the default condition in this release.

In release 02.1.00, ACL CAM sharing enables you to conserve CAM. If this feature is enabled globally, you can share CAM space that is allocated for inbound ACLs between instances on ports that share the same packet processor (PPCR). For example, if you have bound- inbound ACL 101 to ports 1/1 and 1/5, the ACL is stored in a single location in CAM and used by both ports. Table 6.3 describes which ports share PPCRs and can participate in ACL CAM sharing.

NOTE: 10 Gbps ports do not share a PPCR with any other ports. Consequently, they have no need of the ACL CAM sharing feature.

Table 6.3: Common ports per PPCR

Module Type	PPCR Number	Ports supported by PPCR
20 x 1G	PPCR 1	1 - 10
	PPCR2	11 - 20
40 x 1G	PPCR 1	1 - 10
	PPCR 2	11 - 20
	PPCR 3	21 - 30
	PPCR 4	31 - 40

Considerations When Implementing This Feature

The following consideration apply when implementing this feature:

- If you enable ACL CAM sharing, ACL statistics will be generated per-PPCR instead of per-port. If you require the statistics per-port granularity for your application, you cannot use this feature.
- This feature is only applicable for inbound IPv4 ACLs, VPNv4 ACLs, Layer-2 ACLs, and Global PBR policies.
- This feature is not applicable for ACL-based rate-limiting, interface-level PBR policies, and IPv6 ACLs.
- This feature cannot be applied to a virtual interface.
- CAM entry matching within this feature is based on the ACL group ID.

Configuring ACL CAM Sharing

When enabled, ACL CAM sharing is applied across all ports in a system. To apply ACL CAM sharing globally on a NetIron IMR 640 router, use the following command:

```
NI IMR640 Router(config)# enable-acl-cam-sharing
```

Syntax: enable-acl-cam-sharing

ACL CAM sharing is disabled by default.

NOTE: After enabling or disabling ACL CAM sharing, you should rebind all of the ACL's using the **ip rebind-acl all** command.

Configuring Numbered and Named ACLs

When you configure ACLs, 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.

- 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. This document refers to this ACL as *numbered 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. This document refers to this ACL type as *named ACL*.

You can configure up to 100 standard numbered IP ACLs and 100 extended numbered IP ACLs. You also can configure up to 100 standard named ACLs and 100 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. (On BigIron Chassis devices with Management 2 or Management 3 modules, the maximum is 2048.)

Configuring Standard Numbered ACLs

This section describes how to configure standard numbered ACLs with numeric IDs.

- For configuration information on named ACLs, see "Configuring Standard or Extended Named ACLs" on page 6-24.
- For configuration information on extended ACLs, see "Configuring Extended Numbered ACLs" on page 6-17.

Standard ACLs permit or deny packets based on source IP address. You can configure up to 99 standard 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 6-9.

USING THE CLI

To configure a standard ACL and apply it to outgoing traffic on port 1/1, enter the following commands.

```
BigIron(config)# access-list 1 deny host 209.157.22.26 log
BigIron(config)# access-list 1 deny 209.157.29.12 log
BigIron(config)# access-list 1 deny host IPHost1 log
BigIron(config)# access-list 1 permit any
BigIron(config)# int eth 1/1
BigIron(config-if-1/1)# ip access-group 1 out
BigIron(config)# write memory
```

The commands in this example configure an ACL to deny packets from three source IP addresses from being forwarded 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.

Standard ACL Syntax

Syntax: [no] access-list <num> deny | permit <source-ip> | <hostname> <wildcard> [log]

or

Syntax: [no] access-list <num> deny | permit <source-ip>/<mask-bits> | <hostname> [log]

Syntax: [no] access-list <num> deny | permit host <source-ip> | <hostname> [log]

Syntax: [no] access-list <num> deny | permit any [log]

Syntax: [no] ip access-group <num> in | out

Syntax: [no] access-list <num> deny | permit host <source-hostname> host <destination-hostname> [log]

NOTE: The host <source-hostname> host <destination-hostname> parameters are available on devices running Enterprise software release 08.0.00. See “Specifying a Host Name in an ACL Statement” on page 6-37.

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).

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 sub-net 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 sub-net lengths) or 209.157.22.0 0.0.0.255 in the startup-config file.

If you enable the software to display IP sub-net 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 sub-net 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 permitted or denied by the access policy. If you use the **log** argument, the ACL entry is sent to the CPU for processing.

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 | out** parameter specifies whether the ACL applies to incoming traffic or outgoing traffic on the interface to which you apply the ACL. You can apply the ACL to an Ethernet port, POS port, or virtual interface. Note that the **out** option is not supported in the rule-based ACL mode.

NOTE: If the ACL is for the inbound traffic direction on 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 Numbered and Named ACLs” on page 6-14.

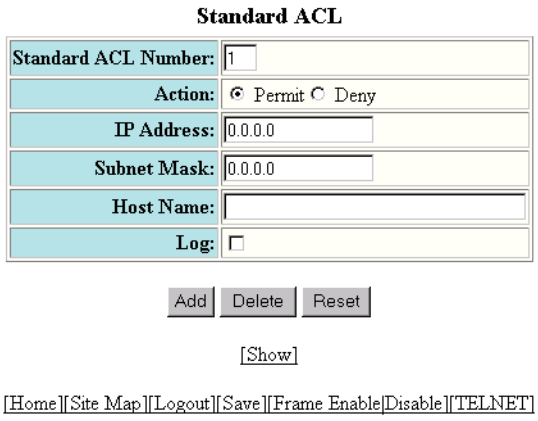
On devices running Enterprise software release 08.0.00 and later, you can specify a hostname for **host** <source-hostname> **host** <destination-hostname> parameter. See “Specifying a Host Name in an ACL Statement” on page 6-37.

USING THE WEB MANAGEMENT INTERFACE

To configure a standard ACL:

1. Log on to the device using a valid user name and password for read-write access. The System configuration dialog is displayed.
2. Click on the plus sign next to Configure in the tree view to display the list of configuration options.
3. Do one of the following to display more configuration options:
   - On a Layer 2 Switch – Click on the plus sign next to System
   - On a Layer 3 Switch – Click on the plus sign next to System or IP. You can access the ACL configuration panels from either location.
4. Select the Standard ACL link.
   - If the device does not already have some standard ACLs, the Standard ACL configuration panel is displayed, as shown in the following example.

   Otherwise, if the device already has some standard ACLs, the Standard ACL table is displayed. This table lists the configured ACLs. Select the Add Standard ACL link to display the Standard ACL configuration panel, as shown in the following example.



5. Change the ACL number in the Standard ACL Number field or use the ACL number displayed in the field.

NOTE: You cannot specify an ACL name.

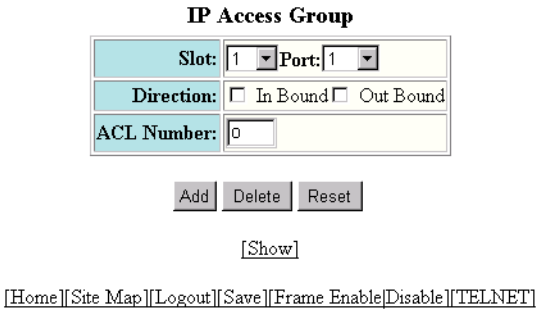
6. Select the ACL action. You can select Permit or Deny:
   - Permit – Forwards traffic or allows management access for the specified IP source.
   - Deny – Drops traffic or denies management access for the specified IP source.

NOTE: If the ACL is a forwarding ACL, the action forwards or drops the traffic. If the ACL is a management access ACL, the action permits or denies management access.

7. Enter the source information. You can enter the source IP address and network mask or the host name.
   - If you enter the address, you also must enter the network mask. To specify “any”, enter “0.0.0.0”.
   - If you enter a host name instead of an IP address, when you click Add to add the ACL, the Web

management interface sends a DNS query for the address. For the query to be successful, the device must have network access to a DNS server and the server must have an Address record for the host. In addition, the device must be configured with a DNS domain name and the IP address of the DNS server.

8. If you specified the Deny action, optionally enable logging by selecting the Log checkbox. If you enable logging for this ACL entry, the software generates Syslog entries for traffic that the ACL denies.
9. Select the IP Access Group link from the tree view.
   - If the device does not already have some ACLs applied to interfaces, the IP Access Group configuration panel is displayed, as shown in the following example.
   - Otherwise, if the device already has some ACLs applied to interfaces, the IP Access Group table is displayed. Select the Add link to display the IP Access Group configuration panel, as shown in the following example.



10. Select the Slot (if you are configuring a Chassis device) and port from the Slot and Port pulldown menus.
11. Specify the traffic direction to which the ACL applies. You can select one or both of the following:
    - In Bound – The ACL applies to traffic received on the port from other devices.
    - Out Bound – The ACL applies to traffic this Foundry device queues for transmission on the port.
12. Enter the ACL number in the ACL Number field.

NOTE: You cannot specify an ACL name.

13. Click the Add button to save the ACL and the association of the ACL with an interface to the device's running-config file.
14. 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.

NOTE: You also can access the dialog for saving configuration changes by clicking on Command in the tree view, then clicking on Save to Flash.

Configuring Extended Numbered ACLs

This section describes how to configure extended numbered ACLs.

- For configuration information on named ACLs, see "Configuring Numbered and Named ACLs" on page 6-14.
- For configuration information on standard ACLs, see "Configuring Standard Numbered ACLs" on page 6-14.

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.

USING THE CLI

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.

```
BigIron(config)# access-list 101 deny tcp host 209.157.22.26 any eq telnet log
BigIron(config)# access-list 101 permit ip any any
BigIron(config)# int eth 1/1
BigIron(config-if-1/1)# ip access-group 101 in
BigIron(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.

```
BigIron(config)# access-list 102 perm icmp 209.157.22.0/24 209.157.21.0/24
BigIron(config)# access-list 102 deny igmp host rkwong 209.157.21.0/24 log
BigIron(config)# access-list 102 deny igrp 209.157.21.0/24 host rkwong log
BigIron(config)# access-list 102 deny ip host 209.157.21.100 host 209.157.22.1 log
BigIron(config)# access-list 102 deny ospf any any log
BigIron(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 and outgoing traffic on port 1/2 and to the incoming traffic on port 4/3.

```
BigIron(config)# int eth 1/2
BigIron(config-if-1/2)# ip access-group 102 in
BigIron(config-if-1/2)# ip access-group 102 out
BigIron(config-if-1/2)# exit
BigIron(config)# int eth 4/3
BigIron(config-if-4/3)# ip access-group 102 in
BigIron(config)# write memory
```

Here is another example of an extended ACL.

```
BigIron(config)# access-list 103 deny tcp 209.157.21.0/24 209.157.22.0/24
BigIron(config)# access-list 103 deny tcp 209.157.21.0/24 eq ftp 209.157.22.0/24
BigIron(config)# access-list 103 deny tcp 209.157.21.0/24 209.157.22.0/24 lt
telnet neq 5
BigIron(config)# access-list 103 deny udp any range 5 6 209.157.22.0/24 range 7 8
BigIron(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 and outgoing traffic on ports 2/1 and 2/2.

```
BigIron(config)# int eth 2/1
BigIron(config-if-2/1)# ip access-group 103 in
BigIron(config-if-2/1)# ip access-group 103 out
BigIron(config-if-2/1)# exit
BigIron(config)# int eth 2/2
BigIron(config-if-2/2)# ip access-group 103 in
BigIron(config-if-2/2)# ip access-group 103 out
BigIron(config)# write memory
```


Extended ACL Syntax

The syntax for configuring extended numbered ACL is different on various Foundry devices.

On the FES X-Series

Use the following for FES X-Series devices:

Syntax: [no] access-list <num> deny | permit <ip-protocol> <source-ip> | <hostname> <wildcard> [<operator> <source-tcp/udp-port>] <destination-ip> | <hostname> [<icmp-type>] <wildcard> [<operator> <destination-tcp/udp-port>] [established] [precedence <name> | <num>] [tos <0 – 63>] [dscp-marking <dscp-value> **802.1p-priority-marking <0 – 7> internal-priority-marking <0 – 7>]** [dscp-marking <dscp-value> **dscp-cos-mapping] [dscp-cos-mapping]**

Syntax: [no] access-list <num> deny | permit host <ip-protocol> any any]

Syntax: [no] ip access-group <num> in

NOTE: The parameters and options that are specific to the FES X-Series are shown above in bold. These are defined in the section “QoS Options for IP ACLs (Rule-Based ACLs)” on page 6-44. All other options and parameters are defined below.

On the NetIron IMR 640

In release 02.0.02 for the NetIron IMR 640, you can match packets for one additional TCP header flag using IPv4 ACLs. The following command implements the additional TCP parameter for IP ACLs.

Syntax: [no] access-list <num> permit | deny tcp any any syn

The <num> parameter indicates the ACL number and must be from 1 - 99 for a standard ACL or from 100 - 199 for and extended ACL

The **tcp** parameter indicates that you are filtering the TCP header.

The **syn** parameter directs the ACL to permit or deny based upon the status of the syn flag in the TCP header. If the contents of the flag is "1" the condition is met.

The syntax presented in “On Other Devices” on page 6-20 also apply to the NetIron IMR 640

On Other Devices

Use the following for IronCore, JetCore, and Terathon devices:

Syntax: [no] access-list <num> deny | permit <ip-protocol> <source-ip> | <hostname> <wildcard> [<operator> <source-tcp/udp-port>] <destination-ip> | <hostname> [<icmp-type> | <icmp-num> | <icmp-type-number> <icmp-code-number>] <wildcard> [<operator> <destination-tcp/udp-port>] [established] [precedence <name> | <num>] [tos <name> | <num>] [ip-pkt-len <value>] [priority 0 | 1 | 2 | 3] [priority-force 0 | 1 | 2 | 3] [priority-mapping <8021p-value>] [dscp-mapping <dscp-value>] [dscp-marking <dscp-value>] [log]

Syntax: [no] access-list <num> deny | permit host <ip-protocol> any any [log]

Syntax: [no] access-list <num> deny | permit host <source-hostname> host <destination-hostname> [log]

NOTE: The host <source-hostname> host <destination-hostname> parameters are available on devices running Enterprise software release 08.0.00. See “Specifying a Host Name in an ACL Statement” on page 6-37.

Syntax: [no] ip access-group <num> in | out

The <num> parameter indicates the ACL number and be from 100 – 199 for an extended ACL.

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. In release 07.6.01 and later, 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 sub-net 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 sub-net lengths) or 209.157.22.0 0.0.0.255 in the startup-config file.

If you enable the software to display IP sub-net 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 sub-net 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. The <icmp-type> parameter is supported in software releases 07.2.06 and later.
- 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 is supported in software releases 07.2.06 and later. This 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
- unreachable

- <num>

On devices running Enterprise IronWare software release 07.8.00, you can enter <icmp-type-number> <icmp-code-number> instead of the <icmp-type> or <icmp-num>. Refer to “ICMP Filtering with Flow-Based ACLs” on page 6-61 for more information on this feature.

The <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**.
- **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.
- **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.

The <tcp/udp-port> parameter specifies the TCP or UDP port number or well-known name. In release 07.6.01 and later, 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 | out** parameter specifies whether the ACL applies to incoming traffic or outgoing traffic on the interface to which you apply the ACL. You can apply the ACL to an Ethernet port, POS port, or a virtual interface.

NOTE: The **out** option is not supported in the rule-based ACL mode.

NOTE: If the ACL is for the inbound traffic direction on 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 Numbered and Named ACLs” on page 6-14.

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 JetCore and 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 **ip-pkt-len** <value> parameter filters ICMP packets based on the IP packet length. The device uses the <value> to match the total length field in the IP header of ICMP packets. You can specify a value from 1 – 65535.

NOTE: This parameter applies only if you specified **icmp** as the <ip-protocol> value. The **ip-pkt-len** <value> parameter is supported in software releases 07.7.00 and later.

The **priority**, **priority-force**, **priority-mapping**, **dscp-mapping**, and **dscp-marking** options are supported in software releases 07.6.01 and later, and apply to JetCore devices and 10 Gigabit Ethernet modules. The **dscp-marking** option is also supported on the FES X-Series. See the section “QoS Options for IP ACLs (Rule-Based ACLs)” on page 6-44.

The **dscp-cos-mapping** option is supported only on the FES X-Series. See the section “QoS Options for IP ACLs (Rule-Based ACLs)” on page 6-44.

The **log** parameter enables SNMP traps and Syslog messages for packets denied by the ACL.

- The **log** parameter is not supported on the FES X-Series devices.
- 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.

On devices running Enterprise software release 08.0.00 and later, you can specify a hostname for **host** <source-hostname> **host** <destination-hostname> parameter. See “Specifying a Host Name in an ACL Statement” on page 6-37.

Configuring Standard or Extended Named ACLs

To configure a named IP ACL, use the following CLI method.

USING THE CLI

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.

The following examples show how to configure a named standard ACL entry and a named extended ACL entry.

Configuration Example for Standard ACL

To configure a named standard ACL entry, enter commands such as the following.

```
BigIron(config)# ip access-list standard Net1
BigIron(config-std-nacl)# deny host 209.157.22.26 log
BigIron(config-std-nacl)# deny 209.157.29.12 log
BigIron(config-std-nacl)# deny host IPHost1 log
BigIron(config-std-nacl)# permit any
BigIron(config-std-nacl)# exit
BigIron(config)# int eth 1/1
BigIron(config-if-1/1)# ip access-group Net1 out
```

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 1/1. 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 6-14.

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 are configuring a named ACL.

Syntax: ip access-list extended | standard <string> | <num>

The **extended | standard** parameter indicates the ACL type.

The <string> parameter is the ACL 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 <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 or 100 – 199 for extended 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 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 Standard Numbered ACLs” on page 6-14.

Configuration Example for Extended ACL

To configure a named extended ACL entry, enter commands such as the following.

```
BigIron(config)# ip access-list extended “block Telnet”
BigIron(config-ext-nacl)# deny tcp host 209.157.22.26 any eq telnet log
BigIron(config-ext-nacl)# permit ip any any
BigIron(config-ext-nacl)# exit
BigIron(config)# int eth 1/1
BigIron(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 6-17.

USING THE WEB MANAGEMENT INTERFACE

You cannot configure IP ACLs using the Web management interface.

Displaying ACL Definitions

To display the ACLs configured on a device, use the **show ip access-lists** command. Here is an example:

```
BigIron(config)# show ip access-lists
Extended IP access list 101
    deny tcp host 209.157.22.26 host 209.157.22.26 eq http log
```

Syntax: show ip access-lists [<num>]

The **show access-list** and **show ip access-list** commands have been updated to display ACL entries with line numbers.

In Enterprise Ironware software release 07.8.00, the **show access-list** and **show ip access-list** commands have been modified to allow you to display a portion of the ACL contents, beginning with a specific entry. For example, if there are five entries in an ACL, you can enter a show command to display its contents beginning with the third entry.

Numbered ACL

For a numbered ACL, you can enter a command such as the following:

```
BigIron(config)# show access-list 99 3
Standard IP access-list 99
3. deny 10.10.10.1
4. deny 192.168.1.13
5. permit any

Total number of entries in ACL 99 5

Grand total number of entries in all ACLs: 200
```

Syntax: show access-list <acl-number> [<line-number>]

Enter the ACL’s number for the <acl-number> parameter.

Determine from which line you want the displayed information to begin and enter that number for the <line-number> parameter.

Named ACL

For a named ACL, enter a command such as the following:

```
BigIron(config)# ip show access-list standard melon 3
Standard IP access-list melon
3. deny host 5.6.7.8
4. deny 192.168.12.3
5. permit any

Total number of entries in ACL melon 5
```



```
Grand total number of entries in all ACLs: 200
```

Syntax: show ip access-list <acl-name> | <acl-number> [<line-number>]

The ACL's name for the <acl-name> parameter or the the ACL's number for <acl-number>.

Determine from which line you want the displayed information to begin and enter that number for the <line-number> parameter.

Displaying of TCP/UDP Numbers in ACLs

Beginning with Enterprise IronWare software release 07.7.00, you can display the port numbers of TCP/UDP application information instead of their TCP/UDP well-known port name in the output of **show** commands and other commands that contain application port information. For example, entering the following command causes the Foundry device to display **80** (the port number) instead of **http** (the well-known port name).

```
BigIron(config)# ip show-acl-service-number
```

Syntax: [no] ip show-acl-service-number

By default, Foundry devices display TCP/UDP application information in named notation.

Displaying ACLs Using Keywords

You limit the displayed ACL entries to those that contain a specified keyword.

Numbered ACL

You may have the following numbered ACL:

```
BigIron(config)# show access-list 99
Standard IP access-list 99
1. deny host 1.2.3.4
2. permit host 5.6.7.8
3. permit host 5.10.11.12
4. permit any

Total number of entries in ACL 99: 4

Grand total number of entries in all ACLs: 200
```

If you want to display ACL entries beginning with the entry that contains the keyword "5" enter the following command:

```
BigIron(config)# show access-list 99 | begin 5
Standard IP access-list 99
2. permit host 5.6.7.8
3. permit host 5.10.11.12
4. permit any

Total number of entries in ACL 99: 4

Grand total number of entries in all ACLs: 200
```

Since the second entry is the first entry that contains the keyword "5", the display begins with line 2.

If you want to display only the ACL entries that contain the keyword "5" enter the following command:

```
BigIron(config)# show access-list 99 | include 5
Standard IP access-list 99
2. permit host 5.6.7.8
3. permit host 5.10.11.12

Total number of entries in ACL 99: 4

Grand total number of entries in all ACLs: 200
```


The second and third entries in the ACL contain the keyword “5” and are displayed in the **show access-list**.

If you want to exclude ACL entries that contain a keyword from the show access-list display, enter the following command:

```
BigIron(config)# show access-list 99 | exclude 5
Standard IP access-list 99
1. deny host 1.2.3.4
4. permit any

Total number of entries in in ACL 99: 4

Grand total number of entries in all ACLs: 200
```

The second and third ACL entries are left out from the display.

Syntax: show access-list <acl-number> **|** begin|exclude|include <keyword>

Enter the ACL’s number for the <acl-number> parameter.

Use the **|** operator to indicate a keyword.

Enter the **begin** <keyword> parameter to start the display beginning with the first line containing the text that matches the keyword. For example, if you enter `begin Total`, the displayed information begins with the line containing the word “Total”.

Enter the **exclude** <keyword> parameter to exclude any lines containing text that match the keyword. For example, if you enter `exclude Total`, any line containing the word “Total” is excluded from the display.

Enter the **include** <keyword> display only those lines containing text that match the keyword. For example, if you enter `include Total`, any line containing the word “Total” is included in the display.

Named ACLs

You may have the following numbered ACL:

```
BigIron(config)# show access-list melon
Standard IP access-list melon
1. deny host 1.2.3.4
2. permit host 5.6.7.8
3. permit host 5.10.11.12
4. permit any

Total number of entries in ACL melon: 4

Grand total number of entries in all ACLs: 200
```

If you want to display ACL entries beginning with the entry that contains the keyword “5” enter the following command:

```
BigIron(config)# show access-list melon | begin 5
Standard IP access-list melon
2. permit host 5.6.7.8
3. permit host 5.10.11.12
4. permit any

Total number of entries in ACL melon: 4

Grand total number of entries in all ACLs: 200
```

Since the second entry is the first entry that contains the keyword “5”, the display begins with line 2.

If you want to display only the ACL entries that contain the keyword “5” enter the following command:

```
BigIron(config)# show access-list melon | include 5
Standard IP access-list melon
2. permit host 5.6.7.8
3. permit host 5.10.11.12
```



```
Total number of entries in ACL melon: 4

Grand total number of entries in all ACLs: 200
```

The second and third entries in the ACL contain the keyword “5” and are displayed in the **show access-list**.

If you want to exclude ACL entries that contain a keyword from the show access-list display, enter the following command:

```
BigIron(config)# show access-list melon | exclude 5
Standard IP access-list melon
1. deny host 1.2.3.4
4. permit any

Total number of entries in in ACL melon: 4

Grand total number of entries in all ACLs: 200
```

The second and third ACL entries are left out from the display.

Syntax: show ip access-list <acl-number> **|** begin | exclude | include <keyword>

Enter the ACL’s number for the <acl-number> parameter.

Use the **|** operator to indicate a keyword.

Enter the **begin** <keyword> parameter to start the display beginning with the first line containing text that matches the keyword. For example, if you enter `begin Total`, the displayed information begins with the line containing the word “Total”.

Enter the **exclude** <keyword> parameter to exclude any lines containing text that match the keyword. For example, if you enter `exclude Total`, any line containing the word “Total” is excluded from the display.

Enter the **include** <keyword> display only those lines containing text that match the keyword. For example, if you enter `include Total`, any line containing the word “Total” is included in the display.

NOTE: If ACL entries, for both numbered and named ACLs, have remarks, the display will also include the remarks if they contain characters that match the specified keywords. For example, ACL 99 might contain the following entries:

```
BigIron(config)# show access-list 99
Standard IP access-list 99
   ACL Remark: Deny Building A
1. deny host 1.2.3.4
   Permit First Floor Users
2. permit host 5.6.7.8
3. deny host 5.10.11.12
4. permit any

Total number of entries in ACL 99: 4

Grand total number of entries in all ACLs: 200
```

To show all entries containing the keyword “deny” you obtain the following output:

```
BigIron(config)# show access-list 99 | include deny
Standard IP access-list 99
   ACL Remark: Deny Building A
1. deny host 1.2.3.4
3. deny host 5.10.11.12

Total number of entries in ACL 99: 4
```



```
Grand total number of entries in all ACLs: 200
```

All lines with the keyword “deny”, including remarks are included in the display.

Modifying ACLs

When you use the Foundry device’s CLI or Web management interface to configure any ACL, the software places the ACL entries in the ACL in the order you enter them. For example, if you enter the following entries in the order shown below, the software always applies the entries to traffic in the same order.

```
BigIron(config)# access-list 1 deny 209.157.22.0/24
BigIron(config)# access-list 1 permit 209.157.22.26
```

Thus, if a packet matches the first ACL entry in this ACL and is therefore denied, the software does not compare the packet to the remaining ACL entries. In this example, packets from host 209.157.22.26 will always be dropped, even though packets from this host match the second entry.

You can use the CLI to reorder entries within an ACL by individually removing the ACL entries and then re-adding them. To use this method, enter “**no**” followed by the command for an ACL entry, and repeat this for each ACL entry in the ACL you want to edit. After removing all the ACL entries from the ACL, re-add them.

This method works well for small ACLs such as the example above, but can be impractical for ACLs containing many entries. Therefore, Foundry devices provide an alternative method. The alternative method lets you upload an ACL list from a TFTP server and replace the ACLs in the device’s running-config file with the uploaded list. Thus, to change an ACL, you can edit the ACL on the file server, then upload the edited ACL to the device. You then can save the changed ACL to the device’s startup-config file.

ACL lists contain only the ACL entries themselves, not the assignments of ACLs to interfaces. You must assign the ACLs on the device itself.

NOTE: The only valid commands that are valid in the ACL list are the **access-list** and **end** commands. The Foundry device ignores other commands in the file.

To modify an ACL by configuring an ACL list on a file server:

1. Use a text editor to create a new text file. When you name the file, use 8.3 format (up to eight characters in the name and up to three characters in the extension).

   NOTE: Make sure the Foundry device has network access to the TFTP server.

2. Optionally, clear the ACL entries from the ACLs you are changing by placing commands such as the following at the top of the file:

   ```
   no access-list 1
   no access-list 101
   ```

   When you load the ACL list into the device, the software adds the ACL entries in the file after any entries that already exist in the same ACLs. Thus, if you intend to entirely replace an ACL, you must use the **no access-list** <num> command to clear the entries from the ACL before the new ones are added.

3. Place the commands to create the ACL entries into the file. The order of the separate ACLs does not matter, but the order of the entries within each ACL is important. The software applies the entries in an ACL in the order they are listed within the ACL. Here is an example of some ACL entries:

   ```
   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 software will apply the entries in ACL 1 in the order shown and stop at the first match. Thus, if a packet is denied by one of the first three entries, the packet will not be permitted by the fourth entry, even if the packet matches the comparison values in this entry.

4. Enter the command "**end**" on a separate line at the end of the file. This command indicates to the software that the entire ACL list has been read from the file.
5. Save the text file.
6. On the Foundry device, enter the following command at the Privileged EXEC level of the CLI:

   copy tftp running-config <tftp-ip-addr> <filename>

   NOTE: This command will be unsuccessful if you place any commands other than **access-list** and **end** (at the end only) in the file. These are the only commands that are valid in a file you load using the **copy tftp running-config…** command.

7. To save the changes to the device's startup-config file, enter the following command at the Privileged EXEC level of the CLI:

   write memory

Here is a complete example of an ACL configuration file.

```
no access-list 1
no access-list 101
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
end
```

NOTE: Do not place other commands in the file. The Foundry device reads only the ACL information in the file and ignores other commands, including **ip access-group** commands. To assign ACLs to interfaces, use the CLI.

Adding, Inserting, Replacing, or Deleting a Comment

You can add, insert, replace, or delete comments to an ACL entry. First enter a show command as discussed in "Displaying a List of ACL Entries" on page 6-33 to determine the line number of the entry you want to update or where you want to insert the new ACL entry. Then enter a command as shown in one of the two sections below.

Numbered ACL: Adding or Replacing a Comment

To add a comment to an ACL entry in a numbered ACL, do the following:

1. use the **show access-list** to display the entries in an ACL. For example:

```
BigIron(config)# show access-list 99
Standard IP access-list 99
1. deny host 1.2.4.5
2. permit host 5.6.7.8
3. permit any

Total number of entries in ACL 99: 3

Grand total number of entries in all ACLs: 200
```

2. To add the comment "Permit all users" to the second entry in the list, enter a command such as the following:

```
BigIron(config)# access-list 99 insert 2 remark Permit all users
```

3. Entering a **show access-list** command displays the following:

```
BigIron(config)# show access-list 99
Standard IP access-list 99
1. deny host 1.2.4.5
   Permit all users
```



```
2. permit host 5.6.7.8
3. permit any

Total number of entries in ACL 99: 3

Grand total number of entries in all ACLs: 200
```

Syntax: access-list <acl-num> insert <line-number> | replace <line-number> remark <comment-text>

Simply entering **access-list** <acl-num> **remark** <comment-text> adds a remark to the next ACL entry you create.

The **insert** <line-number> parameter indicates into which entry the comment is to be added.

The **replace** <line-number> parameter indicates which entry's remark will be replaced.

The **remark** <comment-text> adds a comment to the ACL entry. The remark can have 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 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.

Complete the syntax by specifying any options you want for the ACL entry. Options you can use to configure standard or extended numbered ACLs are discussed in the *Foundry Enterprise Configuration and Management Guide*.

Numbered ACLs: Deleting a Comment

To delete a remark from a named ACL entry, enter the following command:

```
BigIron(config)# access-list 99 delete 2 remark
```

Syntax: delete <line-number> remark

Named ACLs: Adding a Comment to a New ACL

You can add, insert, replace, and delete ACL entry remarks. To add a comment, do the following:

1. Use the **show access-list** command to display the contents of the ACL. For example, you may have an ACL named "melon" and a **show access-list** command shows that it has only one entry.

```
BigIron(config)# show access-list melon
Standard IP access-list 99
1. deny host 1.2.4.5
```

2. Add a new entry with a remark to this named ACL by entering commands such as the following:

```
BigIron(config)# ip access-list standard melon
BigIron(config-std-nacl)# remark Deny traffic from Marketing
BigIron(config-std-nacl)# deny 5.6.7.8
```

3. Enter a **show access-list** command displays the new ACL entry with its remark:

```
BigIron(config)# show access-list melon
Standard IP access-list melon
1. deny host 1.2.4.5
   Deny traffic from Marketing
2. permit host 5.6.7.8

Total number of entries in ACL melon: 2

Grand total number of entries in all ACLs: 200
```

Syntax: ip access-list standard | extended <acl-name> | <acl-num>

Syntax: remark <comment-text>

Syntax: deny <options> | permit <options>

The **standard** | **extended** parameter indicates the ACL type.

The <acl-name> parameter is the ACL 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, enter a number from 1 – 99 for standard ACLs or 100 – 199 for extended ACLs.

The **remark** <comment-text> adds a comment to the ACL entry that you are about to create. The comment can have 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 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.

Enter **deny** to deny the specified traffic or **permit** to allow the specified traffic. Complete the configuration by specifying <options> for the standard or extended ACL entry. Options you can use to configure standard or extended named ACLs are discussed in the *Foundry Enterprise Configuration and Management Guide*.

Named ACLs: Inserting or Replacing Comments to Existing ACL Entries

To insert a comment to an existing entry in the ACL named melon, or to replace a comment for an ACL entry, display the list of entries in the ACL.

```
BigIron(config)# show access-list melon
Standard IP access-list melon
1. deny host 1.2.4.5
2. permit host 5.6.7.8
3. permit any

Total number of entries in ACL melon: 3

Grand total number of entries in all ACLs: 200
```

To add the comment "Permit all users" to the second entry in the list, enter a command such as the following:

```
BigIron(config)# ip access-list standard melon
BigIron(config-std-nacl)# insert 3 remark Permit all users
```

Use the **show access-list** command to display the updated ACL:

```
BigIron(config)# show access-list melon
Standard IP access-list melon
1. deny host 1.2.4.5
2. permit host 5.6.7.8
   Permit all users
3. permit ip any any

Total number of entries in ACL melon: 3

Grand total number of entries in all ACLs: 200
```

To replace the comment for the third entry, enter commands such as the following:

```
BigIron(config)# ip access-list standard melon
BigIron(config-std-nacl)# replace 3 remark All users allowed
```

Entering the **show access-list** command displays the updated ACL:

```
BigIron(config)# show access-list melon
Standard IP access-list melon
1. deny host 1.2.4.5
2. permit host 5.6.7.8
   All users allowed
3. permit ip any any

Total number of entries in ACL melon: 3
```



```
Grand total number of entries in all ACLs: 200
```

Syntax: ip access-list standard | extended <acl-name> | <acl-num>

Syntax: insert <line-number> | replace <line-number> remark <comment-text>

The **standard | extended** parameter indicates the ACL type.

The <acl-name> parameter is the ACL 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. You can specify a number from 1 – 99 for standard ACLs or 100 – 199 for extended ACLs.

The **insert** <line-number> parameter indicates into which entry the comment is to be added. The **replace** <line-number> command indicates which remarks will be replaced.

The **remark** <comment-text> adds a comment to the ACL entry. The remark can have 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.

Complete the syntax by specifying options for the ACL entry. Options you can use to configure standard or extended named ACLs are discussed in the *Foundry Enterprise Configuration and Management Guide*.

Deleting a Remark from a Named ACL.

To delete a remark from a named ACL, enter the following command:

```
BigIron(config)# ip access-list standard melon
BigIron(config-std-nacl)# delete 3 remark
```

Syntax: delete <line-number> remark

Inserting, Deleting, and Replacing ACL Entries

ACL support in most Foundry devices append newly created ACL entries to the end of the ACL list. Since ACL entries are applied to data packets in the order they appear in a list, you needed to create ACLs in the order you want them applied.

In devices running Enterprise IronWare software release 07.8.00 and later, the CLI command for standard and extended ACLs has been enhanced with the following options:

- Insert a new ACL entry within an existing ACL
- Delete an entry from an ACL
- Replace an existing ACL entry

Displaying a List of ACL Entries

The **show access-list** and **show ip access-list** commands displays ACL entries with line numbers.

Numbered ACLs

To display the contents of numbered ACLs, enter a command such as the following:

```
BigIron# show access-list 99
Standard IP access list 99
1. deny host 1.2.4.5
2. deny host 5.6.7.8
3. permit any

Total number of entries in ACL 99: 3

Grand total number of entries in all ACLs: 200
```

Syntax: show access-list <acl-num> | all

Named ACLs

To display the contents of named ACLs, enter a command such as the following:

```
BigIron# show ip access-list melon
Standard IP access list melon
1. deny host 1.2.4.5
2. deny host 5.6.7.8
3. permit any

Total number of entries in ACL melon: 3

Grand total number of entries in all ACLs: 200
```

Syntax: show ip access-list <acl-num> | <acl-name>

Inserting an ACL Entry

To insert an entry in an ACL, use a show command to determine where you want to insert the new entry. Then enter commands as shown in the following sections.

Numbered ACLs

```
BigIron(config)# access-list 99 insert 2 deny 5.6.7.8
```

The example inserts a new ACL entry in line 2 of ACL 99.

Syntax: access-list <acl-num> insert <line-num>

The <acl-num> parameter identifies the numbered ACL into which the new entry will be inserted. This number can be from 1 - 99 for standard ACLs or from 100 - 199 for extended ACLs.

The **insert** <line-num> parameter indicates where the new ACL entry will be inserted. After the new entry is inserted, the ACL list is renumbered.

Complete the configuration by specifying options for the ACL entry. Options you can use to configure standard or extended numbered ACLs are discussed in the *Foundry Enterprise Configuration and Management Guide*.

Named ACLs

To insert an ACL entry that denies host 10.1.1.1 as the second entry in an ACL named melon, use the **show ip access-list** to display the current entries in melon:

```
BigIron(config)# show ip access-list standard melon
Standard IP access-list melon
1. deny host 1.2.4.5
2. deny host 5.6.7.8
3. permit any

Total number of entries in ACL melon: 3

Grand total number of entries in all ACLs: 200
```

To add an entry that denies IP Host 10.1.1.1 from any source at line 2, enter:

```
BigIron(config)# ip access-list standard melon
BigIron(config-std-nacl)# insert 2 deny host 10.1.1.1
```

Entering a **show ip access-list melon** command displays the updated list.

```
BigIron(config)# ip show access-list standard melon
Standard IP access-list melon
1. deny host 1.2.4.5
2. deny host 10.1.1.1
3. deny host 5.6.7.8
4. permit any
```



```
Total number of entries in ACL melon: 4

Grand total number of entries in all ACLs: 200
```

The updated list has been renumbered.

Syntax: insert <line-num> deny <options>

Syntax: insert <line-number> permit <options>

Syntax: insert <line-number> remark <comment-text>

The **standard | extended** parameter indicates the ACL type.

The <acl-name> parameter is the ACL 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. You can specify a number from 1 – 99 for standard ACLs or 100 – 199 for extended ACLs.

Enter the number of the ACL for <acl-num>.

ACL entry options for ip access-list are configured under the standard or extended ACL level.

The **insert** <line-num> parameter indicates where on the list the ACL entry is to be inserted.

Enter **deny** if you are creating an entry that denies access to the device. Enter **permit** to create an entry that allows access to a device. Enter a **remark** to add a remark to an ACL entry.

Complete the configuration by specifying <options> for standard or extended ACL entries. Options you can use to configure standard or extended named ACLs are discussed in the *Foundry Enterprise Configuration and Management Guide*.

If you entered **remark**, enter a text string for <comment-text>.

Deleting an ACL Entry From Within a List

If you want to delete an ACL entry from within a list, enter a show command as discussed in “Displaying a List of ACL Entries” on page 6-33 to determine the line number of the entry you want to delete. Then enter a command as shown one of the two sections below.

Numbered ACLs

If you want to delete the second entry from a numbered ACL such as ACL 99, display the contents of the list.

```
BigIron(config)# show access-list 99
Standard IP access-list 99
1. deny host 1.2.4.5
2. deny host 5.6.7.8
3. permit any

Total number of entries in ACL 99: 3

Grand total number of entries in all ACLs: 200
```

Enter the following command:

```
BigIron(config)# access-list 99 delete 2
```

Display the contents of the updated list:

```
BigIron(config)# show ip access-list 99
Standard IP access-list 99
1. deny host 1.2.4.5
2. permit any

Total number of entries in ACL 99: 2

Grand total number of entries in all ACLs: 200
```


The updated list has been renumbered.

Syntax: access-list <acl-number> delete <line-number>

The <line-number> parameter specifies the ACL entry to be deleted. The <acl-num> parameter allows you to specify an ACL number if you prefer. If you specify a number, enter a number from 1 – 99 for standard ACLs or 100 – 199 for extended ACLs.

The **delete** <line-num> parameter indicates the ACL entry to be deleted.

Complete the configuration by specifying options for the ACL entry. Options you can use to configure standard or extended numbered ACLs are discussed in “Configuring Standard Numbered ACLs” on page 6-14 and “Configuring Extended Numbered ACLs” on page 6-17.

Named ACLs

To delete an ACL entry from an ACL named "melon", enter the following command to display the contents of the ACL list:

```
BigIron#show access-list melon
Standard IP access list melon
1. deny host 1.2.4.5
2. deny host 10.1.1.1
3. deny host 5.6.7.8
4. permit any

Total number of entries in ACL melon: 4

Grand total number of entries in all ACLs: 200
```

To delete the second ACL entry from the list, enter a command such as the following :

```
BigIron(config)# ip access-list standard melon
BigIron(config-std-nacl)# delete 2
```

Enter the **ip show access-list melon** command to display the updated list.

```
BigIron(config)# ip show access melon all
Standard IP access list melon
1. deny host1.2.4.5
2. deny host 5.6.7.8
3. permit any

Total number of entries in ACL melon: 3

Grand total number of entries in all ACLs: 200
```

The updated list has been renumbered.

Syntax: ip access-list standard | extended <acl-name> | <acl-number>

Syntax: delete <line-number>

The **extended | standard** parameter indicates the ACL type.

The <acl-name> parameter is the ACL 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, enter a number from 1 – 99 for standard ACLs or 100 – 199 for extended ACLs.

The delete parameter is entered in the standard or extended ip access-list level.

The **delete** <line-number> parameter indicates first ACL entry to be deleted.

Specifying a Host Name in an ACL Statement

On device running Enterprise software release 08.0.00 and later, you can specify a host name within the configuration of an access control list. For example, the following are valid ACL statements in release 08.0.00:

```
access-list 101 deny ip host www.google.com host www.ibm.com
access-list 102 permit tcp host www.sbc.com any eq telnet log
```

This enhancement is valid for both rule-based and flow-based ACLs.

When you enter an ACL statement that contains a host name in the CLI, the Foundry device attempts to resolve the host name using the DNS resolver. The Foundry device checks its DNS cache for an entry corresponding to the host name; if an entry is not found, the device sends a DNS query to the configured DNS server. When the host name is resolved to an IP address, a Layer 4 CAM entry is created for the ACL.

If the host name cannot be resolved, then the ACL statement is not activated. When you enter the **show run** command, a line such as the following appears for ACL statements referring to hosts that aren't resolved:

```
permit udp host 3.3.3.3 host www.yahoo.com (not in effect)
```

When a host name in an ACL statement cannot be resolved, the Foundry device will periodically attempt to resolve it. In addition, the Foundry device will periodically attempt to resolve host names that had been resolved previously, but are no longer resolved because TTL expired. If the resolution is successful and the IP address has changed, then the ACL's Layer 4 CAM entry is updated; otherwise, it remains unchanged.

Notes

- In order for the host name ACL feature to work, DNS must be properly configured on the Foundry device, and the configured DNS server must be reachable from the Foundry device.
- If the host name times out, the Foundry device attempts to resolve the host name again. If the resolution is successful, there are two possibilities:
  - When the host-name-to-IP-address mapping is unchanged, the Layer 4 CAM entry remains intact.
  - When the host-name-to-IP-address mapping changes, the Layer 4 CAM entry is flushed with the new IP address for the host.

  If the host name resolution is unsuccessful, the Layer 4 CAM entry for the ACL is removed. The output of the **show run** command displays the ACL entry as "not in effect". For example, if the host name www.foundrynet.com is not resolved, then the output of the **show run** command would have the following line:

  ```
  permit udp host 3.3.3.3 host www.foundrynet.com (not in effect)
  ```

  The Foundry device will then make multiple attempts to resolve the host name. If and when the host name is resolved, a new Layer 4 CAM entry is created for the newly resolved IP address.
- When the Foundry device is booted, it may take a few seconds for an ACL statement containing a host name to take effect. This is because the device must first resolve the host name and create a Layer 4 CAM entry. During this brief period, the output of the **show run** command displays the ACL entry as "not in effect".

Replacing an ACL Entry

If you want to replace the definition of an ACL entry, enter a show command as discussed in "Displaying a List of ACL Entries" on page 6-33 to determine the line number of the entry you want to update. Then enter a command as shown in one of the two sections below.

Numbered ACLs

For example, display the ACL entries in ACL 99.

```
BigIron(config)# show access-list 99
Standard IP access-list 99
1. deny host 1.2.4.5
2. deny host 10.8.9.1
3. permit any
```



```
Total number of entries in ACL 99: 3

Grand total number of entries in all ACLs: 200
```

If you want to modify the second entry in ACL 99, enter a command such as the following:

```
BigIron(config)# access-list 99 replace 2 permit 5.6.7.8
```

Entering a show access-list command displays the updated list.

```
BigIron(config)# show access-list 99
Standard IP access-list 99
1. deny host 1.2.4.5
2. permit host 5.6.7.8
3. permit any

Total number of entries in ACL 99: 3

Grand total number of entries in all ACLs: 200
```

Syntax: access-list <acl-num> replace <line-num>

The <acl-num> parameter identifies the numbered ACL from which the ACL entry will be inserted. This number can be from 1 - 99 for standard ACLs or 100 - 199 for extended ACLs.

The **replace** <line-num> parameter indicates which line you want to replace in the current ACL list. After the new entry is inserted, the ACL list is renumbered.

Complete the configuration by specifying options for the ACL entry. Options you can use to configure standard or extended numbered ACLs are discussed in the *Foundry Enterprise Configuration and Management Guide*.

Named ACLs

To replace an entry in the ACL named melon, display list of entries in the ACL.

```
BigIron(config)# show ip access-list melon
Standard IP access-list melon
1. deny ip host 1.2.4.5
2. deny 10.1.1.1
3. permit any

Total number of entries in ACL melon: 3

Grand total number of entries in all ACLs: 200
```

If you want to modify the second entry in ACL melon, enter a command such as the following:

```
BigIron(config)# ip access-list standard melon
BigIron(config-std-nacl)# replace 2 permit 5.6.7.8
```

Entering a show ip access-list command displays the updated list.

```
BigIron(config)# show ip access-list melon
Standard IP access-list melon
1. ip host 1.2.4.5
2. permit ip host 5.6.7.8
3. permit any

Total number of entries in ACL melon: 3

Grand total number of entries in all ACLs: 200
```

Syntax: ip access-list standard | extended <acl-name> | <acl-num> replace <line-num>

Syntax: replace <line-number> deny <options>

Syntax: replace <line-number> permit <options>

Syntax: replace <line-number> remark <comment-text>

The **standard | extended** parameter indicates the ACL type.

The <acl-name> parameter is the ACL 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, enter a number from 1 – 99 for standard ACLs or 100 – 199 for extended ACLs.

Enter the number of the ACL for <acl-num>.

ACL entry options for ip access-list are configured under the standard or extended ACL level.

The **insert** <line-num> parameter indicates where on the list the ACL entry is to be inserted.

Enter **deny** if you are creating an entry that denies access to the device. Enter **permit** to create an entry that allows access to the device. Enter a **remark** to add a remark to an ACL entry.

Complete the configuration by specifying <options> for the standard or extended ACL entry. Options you can use to configure standard or extended named ACLs are discussed in the *Foundry Enterprise Configuration and Management Guide*.

If you entered **remark**, enter a text string for <comment-text>.

Applying an ACLs to Interfaces

Configuration examples in the section “Configuring Numbered and Named ACLs” on page 6-14 show that you apply ACLs to interfaces using the **ip access-group** command. This section present additional information about applying ACLs to interfaces.

Reapplying Modified ACLs

For flow-based and rule-based ACLs, if you make an ACL configuration change, you must reapply the ACLs to their interfaces to place the change into effect.

NOTE: On devices running Enterprise software releases, this section applies to software release 07.6.01 and later. This section also applies to other Foundry devices.

An ACL configuration change includes any of the following:

- Adding, changing, or removing an ACL or an entry in an ACL
- Changing a PBR policy
- Enabling or disabling the TCP strict mode or UDP strict mode (flow-based ACLs only)
- Changing JetCore ToS-based QoS mappings (since JetCore QoS uses the Layer 4 CAM)

To reapply ACLs following an ACL configuration change, enter the following command at the global CONFIG level of the CLI:

```
BigIron(config)# ip rebind-acl all
```

Syntax: [no] ip rebind-acl <num> | <name> | all

Applying an ACL to Outgoing Traffic on a Port

NOTE: This feature applies to release 02.0.02 for the NetIron IMR 640 only.

Release 02.0.02 for the NetIron IMR 640 supports filtering of both inbound and outbound IPv4 traffic.

The syntax for outbound ACL support on the NetIron IMR 640 is the same as on a NetIron Chassis device. However, outbound ACL support on the NetIron IMR 640 is implemented in hardware, making it possible for the NetIron IMR 640 to filter traffic at line-rate speed on 10 Gigabit interfaces.

To apply an ACL to outgoing traffic on a port, enter commands such as the following:


```
NI IMR640 Router(config)# int eth 1/1
NI IMR640 Router(config-if-1/1)# ip access-group 1 out
```

The above command applies ACL 1 to outgoing traffic on port e 1/1.

Syntax: [no] ip access-group <num> in | out

Applying ACLs to a Virtual Routing Interface

NOTE: This feature is not supported on the FastIron Edge Switch.

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. If the ACL is for the inbound traffic direction, 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: This feature applies only to a virtual interface's inbound direction. You cannot use this feature to specify a subset of ports for a virtual interface's outbound direction.

To apply an ACL to a subset of ports within a virtual interface, enter commands such as the following:

```
BigIron(config)# vlan 10 name IP-subnet-vlan
BigIron(config-vlan-10)# untag ethernet 1/1 to 2/12
BigIron(config-vlan-10)# router-interface ve 1
BigIron(config-vlan-10)# exit
BigIron(config)# access-list 1 deny host 209.157.22.26 log
BigIron(config)# access-list 1 deny 209.157.29.12 log
BigIron(config)# access-list 1 deny host IPHost1 log
BigIron(config)# access-list 1 permit any
BigIron(config)# interface ve 1
BigIron(config-vif-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 <num> in ethernet <portnum> [<portnum>...] to <portnum>

ACL Logging

You may want the software to log entries for ACLs in the syslog. This section present the how logging is processed by flow-based and rule-based ACLs.

ACL Logging for Flow-Based ACLs

ACL logging is disabled by default for flow-based ACLs. However, when you configure an ACL entry, you can enable logging for that entry by adding the **log** parameter to the end of the CLI command for the entry.

When you enable logging for an ACL entry, statistics for packets that match the permit or deny conditions of the ACL entry 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 Foundry device's Syslog buffer and in SNMP traps sent by the device.

The first time an ACL entry permits or denies a packet, the software immediately generates a Syslog entry and 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 has denied a packet. The message indicates the number of packets denied by the ACL entry during the previous five minutes.

If no ACL entries explicitly permit or deny packets during an entire five-minute timer interval, the timer stops. The timer restarts when an ACL entry explicitly permits or denies a packet.

NOTE: The timer for logging packets denied by Layer 2 filters is separate.

The software generates log entries only when packets are explicitly permitted or explicitly denied by ACLs. The software does not generate log entries for implicitly permitted or denied entries. Depending on how many entries have the log option and how often packets match those entries, ACL performance can be affected. Use the **log** option only when needed.

Configuring the Layer 4 Session Log Timer

In Enterprise releases 07.6.03 and later, you can configure the Layer 4 session log timer, which is used for keeping track of packets explicitly denied by an ACL.

When you enable logging for an ACL entry, statistics for packets that match the permit or deny conditions of the ACL entry are logged in the Foundry device's Syslog buffer and in SNMP traps sent by the device. The first time an ACL entry permits or denies a packet, the software immediately generates a Syslog entry and SNMP trap. The software also starts the Layer 4 session log timer. The timer keeps track of all packets explicitly denied by the ACL entries. When the timer expires, the software generates a single Syslog entry for each ACL entry that has denied a packet. The message indicates the number of packets denied by the ACL entry from the time that the timer was started. If no ACL entries explicitly permit or deny packets during an entire timer interval, the timer stops. The timer restarts when an ACL entry explicitly permits or denies a packet.

To store information about denied packets during the timer interval, the device makes entries in its Layer 4 session table. If a large number of packets are denied by the ACL during the timer interval, it can consume a large portion of the device's Layer 4 resources. To prevent this from happening, starting in release 07.6.03, you can configure the timer interval to be a shorter length of time. In releases prior to 07.6.03, the timer interval was set to 5 minutes and was not configurable.

For example, to set the timer interval to 2 minutes, enter the following command:

```
BigIron(config)# ip access-list logging-age 2
```

Syntax: ip access-list logging-age <minutes>

You can set the timer to between 1 and 10 minutes. The default is 5 minutes.

ACL Logging for Rule-Based ACLs

Rule-based ACLs do not support the **log** option. Even when rule-based ACLs are enabled, if an ACL entry has the **log** option, traffic that matches that ACL is sent to the CPU for processing. Depending on how many entries have the log option and how often packets match those entries, ACL performance can be affected.

If your configuration already contains ACLs that you want to use with rule-based ACLs, but some of the ACLs contain the **log** option, the software changes the ACL mode to flow-based for the traffic flows that match the ACL. Changing the mode to flow-based enables the device to send the matching flows to the CPU for processing. This is required because the CPU is needed to generate the Syslog message.

You can globally disable ACL logging without the need to remove the **log** option from each ACL entry. When you globally disable ACL logging, the ACL entries remain unchanged but the **log** option is ignored and the ACL can use the rule-based ACL mode. This enables you to use the ACLs in the rule-based ACL mode. You also can configure the device to copy traffic that is denied by a rule-based ACL to an interface. This option allows you to monitor the denied traffic without sending the traffic to the CPU.

To globally disable ACL logging, enter the following command at the global CONFIG level of the CLI:

```
BigIron(config)# ip access-list disable-log-to-cpu
```

Syntax: [no] ip access-list disable-log-to-cpu

To re-enable ACL logging, enter the following command:

```
BigIron(config)# no ip access-list disable-log-to-cpu
```


Syslog Message for Changed ACL Mode

If the device changes the ACL mode from rule-based to flow-based, the device generates one of the following Syslog notification messages:

- ACL insufficient L4 session resource, using flow based ACL instead.
- ACL exceed max DMA L4 cam resource, using flow based ACL instead. See “Specifying the Maximum Number of CAM Entries for ACLs (Rule-Based ACLs)” on page 6-12.
- ACL insufficient L4 cam resource, using flow based ACL instead.

For explanation of the messages, see Table A.2 in Appendix A of the *Foundry Switch and Router Installation and Basic Configuration Guide*.

Copying Denied Traffic to a Mirror Port for Monitoring

Although rule-based ACLs do not support ACL logging, you nonetheless can monitor the traffic denied by rule-based ACLs. To do so, attach a protocol analyzer to a port and enable the device to redirect traffic denied by ACLs to that port.

To redirect traffic denied by ACLs, enter the following command at the interface configuration level:

```
BigIron(config-if-1/1)# ip access-group redirect-deny-to-interf
```

Syntax: [no] ip access-group redirect-deny-to-interf

Enter the command on the port to which you want the denied traffic to be copied.

NOTE: The software requires that an ACL has already been applied to the interface.

When you enable redirection, the deny action of the ACL entry is still honored. Traffic that matches the ACL is not forwarded.

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 one to ten minutes, depending on the value of the timer interval. 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. For more information about the timer, see “Configuring the Layer 4 Session Log Timer” on page 6-41.

NOTE: For an ACL entry to be eligible to generate a Syslog entry for permitted or 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, use one of the following methods.

USING THE CLI

Enter the following command from any CLI prompt:

```
BigIron(config)# show log
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

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)
00d07h03m30s:warning:list 101 denied tcp 209.157.22.26(0)(Ethernet 4/18
0010.5a1f.77ed) -> 198.99.4.69(http), 1 event(s)
00d06h58m30s:warning:list 101 denied tcp 209.157.22.198(0)(Ethernet 4/18
0010.5a1f.77ed) -> 198.99.4.69(http), 1 event(s)
```

In this example, the two-line message at the bottom is the first entry, which the software immediately generates the first time an ACL entry permits or denies a packet. In this case, an entry in ACL 101 denied a packet. The packet was a TCP packet from host 209.157.22.198 and was destined for TCP port 80 (HTTP) on host 198.99.4.69.

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 denied packets.

In this example, the software generates the second log entry five minutes later.

The time stamp for the third entry is much later than the time stamps for the first two entries. In this case, no ACLs denied packets for a very long time. In fact, since no ACLs denied packets during the five-minute interval following the second entry, the software stopped the ACL log timer. The software generated the third entry as soon as the ACL denied a packet. The software restarted the five-minute ACL log timer at the same time. As long as at least one ACL entry permits or denies a packet, the timer continues to generate new log entries and SNMP traps every five minutes.

You can also configure the maximum number of ACL-related log entries that can be added to the system log over a one-minute period. For example, to limit the device to 100 ACL-related syslog entries per minute:

```
BigIron(config)# max-acl-log-num 100
```

Syntax: [no] max-acl-log-num <num>

You can specify a number between 0 – 4096. The default is 256. Specifying 0 disables all ACL logging.

USING THE WEB MANAGEMENT INTERFACE

1. Log on to the Layer 3 Switch using a valid user name and password for read-write access. The System configuration panel is displayed.
2. Click on the plus sign next to Monitor in the tree view to expand the list of configuration options.
3. Select the System Log link.

Displaying ACL Statistics for Flow-Based ACLs

To display ACL statistics for flow-based ACLs, enter the following command:

```
BigIron(config)# show ip acl-traffic
```



```
ICMP inbound packets received 400
ICMP inbound packets permitted 200
ICMP inbound packets denied 200
```

Syntax: show ip acl-traffic

The command lists a separate set of statistics for each of the following IP protocols:

- ICMP
- IGMP
- IGRP
- IP
- OSPF
- TCP
- UDP
- Protocol number, if an ACL is configured for a protocol not listed above

For TCP and UDP, a separate set of statistics is listed for each application port.

Clearing Flow-Based ACL Statistics

To clear the ACL statistics, enter the following command at the Privileged EXEC level of the CLI:

```
BigIron(config)# clear ip acl-traffic
```

Syntax: clear ip acl-traffic

QoS Options for IP ACLs (Rule-Based ACLs)

NOTE: QoS options for IP ACLs are supported in software releases 07.6.01 and later, as well as in release 02.0.02 for the NetIron IMR 640 (see the Note about the specific QoS options for the NetIron IMR 640). DSCP marking and DSCP CoS mapping are supported on the FES X-Series running software release 01.0.00 or later.

QoS options enable you to perform QoS for packets that match the ACLs. Using an ACL to perform QoS is an alternative to the following methods:

- 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 2-11.)
- Enabling the IP ToS-based QoS feature described in “Configuring Enhanced Quality of Service” on page 4-1.

NOTE: If you use an ACL on an interface, ToS-based QoS assumes that the ACLs will perform QoS for all packets except the packets that match the **permit ip any any** ACL.

The following QoS ACL options are supported:

- **priority** – Assigns traffic that matches the ACL to a hardware forwarding queue. In addition to changing the internal forwarding priority, if the outgoing interface is an 802.1q interface, this option maps the specified priority to its equivalent 802.1p (CoS) priority and marks the packet with the new 802.1p priority.
- **internal-priority-marking** and **802.1p-priority-marking** – Supported on the FES X-Series. These commands are similar to the **priority** command (described above) in software releases 07.6.01 and later. 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**).
- **priority-force** – Assigns packets of outgoing traffic that match the ACL to a specific hardware forwarding queue, even though the incoming packet may be assigned to another queue. Specify one of the following QoS queues:

  - 0 – qosp0
  - 1 – qosp1
  - 2 – qosp2
  - 3 – qosp3
- **priority-mapping** – Matches on the packet's 802.1p value. This option does not change the packet's forwarding priority through the device or mark the packet.

  NOTE: This option is not supported on 10 Gigabit Ethernet modules.

- **dscp-mapping** – Matches on the packet's DSCP value. This option does not change the packet's forwarding priority through the device or mark the packet.
- **dscp-marking** – Marks the DSCP value in the outgoing packet with the value you specify.
- **dscp-cos-mapping** – Supported on the FES X-Series. This option is similar to the **dscp-mapping** command (described earlier) in software releases 07.6.01 and later. 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.

NOTE: In release 02.0.02, the NetIron IMR 640 supports the **priority**, **priority-force**, and **priority-mapping** QoS options.

For the **priority-force** option, if a packet's 802.1p value is forced to another value by its assignment to a lower value queue, it will retain that value when it is sent out through the outbound port.

The default behavior on previous revisions of this feature was to send the packet out with the the higher of two possible values: the initial 802.1p value that the packet arrived with or the new (higher) priority that the packet has been "forced" to. To return operation to the previous default behavior, see the merge-egress priorities command in "Configuring Egress Priority Merging" on page 10-8.

Using an ACL to Change the Forwarding Queue

On the FES X Series

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 6.4 lists the default mappings of hardware forwarding queues to 802.1p priorities on the FES X-Series.

Table 6.4: FES X-Series 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 **802.1p-priority-marking** <0 – 7> parameter re-marks the packets of the 802.1q traffic that matches the ACL with this new 802.1p priority, or marks the packets of the non-802.1q traffic that matches the ACL with this 802.1p priority, later at the outgoing 802.1q interface.

Syntax: ...[dscp-marking <dscp-value> **802.1p-priority-marking** <0 – 7> **internal-priority-marking** <0 – 7>]

NOTE: For complete syntax information, see “Extended ACL Syntax” on page 6-20.

On Other Devices

The **priority** option enables you to assign traffic that matches the ACL to a specific hardware forwarding queue (qosp0, qosp1, qosp2, or qosp3).

In addition to changing the internal forwarding priority, if the outgoing interface is an 802.1q interface, this option maps the specified priority to its equivalent 802.1p (CoS) priority and marks the packet with the new 802.1p priority. Table 6.5 lists the default mappings of hardware forwarding queues to 802.1p priorities.

Table 6.5: Default Mapping of Forwarding Queues to 802.1p Priorities

Forwarding Queue	qosp0	qosp0	qosp1	qosp1	qosp2	qosp2	qosp3	qosp3
802.1p	0	1	2	3	4	5	6	7

Here is an example of how to use the **priority** option.

```
BigIron(config)# access-list 110 permit ip any any priority 2
BigIron(config)# interface 1/1
BigIron(config-if-1/1)# ip access-group 110 out
```

These commands configure an extended ACL that places all IP traffic that is queued to be sent on port 1/1 into hardware forwarding queue qosp2 on that port. In addition, if port 1/1 is tagged, the ACL also marks the packets' 802.1p value.

The **priority 0 | 1 | 2 | 3** parameter specifies the QoS queue:

- 0 – qosp0
- 1 – qosp1
- 2 – qosp2
- 3 – qosp3

NOTE: This **priority** option provides the same function as the Layer 4 IP access policies supported on BigIron Chassis devices. If you configure both a Layer 4 IP access policy and an extended ACL to set the hardware forwarding priority for the same traffic, the device uses the ACL instead of the IP access policy.

The **priority-force** parameter allows you assign packets of outgoing traffic that match the ACL to a specific hardware forwarding queue, even though the incoming packet may be assigned to another queue. Select one of the following QoS queue:

- 0 – qosp0
- 1 – qosp1
- 2 – qosp2
- 3 – qosp3

Matching on a Packet's 802.1p Value

The **priority-mapping** option matches on the packet's 802.1p value. This option does not change the packet's forwarding priority through the device or mark the packet.

NOTE: This option is not supported on 10 Gigabit Ethernet modules.

To configure an ACL that matches on a packet's 802.1p priority, enter a command such as the following:


```
BigIron(config)# access-list 111 permit ip 1.1.1.0 0.0.0.255 2.2.2.x 0.0.0.255
priority-mapping 0
```

Syntax: ...**priority-mapping** <8021p-value>

NOTE: For complete syntax information, see “Extended ACL Syntax” on page 6-20.

Matching on a Packet’s DSCP Value

On the FES X-Series

The **dscp-cos-mapping** option on the FES X-Series 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.

Syntax: ...[dscp-marking <dscp-value> **dscp-cos-mapping**]

OR

Syntax: ...[dscp-cos-mapping]

NOTE: For complete syntax information, see “Extended ACL Syntax” on page 6-20.

On Other Devices

The **dscp-mapping** 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:

```
BigIron(config)# access-list 112 permit ip 1.1.1.0 0.0.0.255 2.2.2.x 0.0.0.255 dscp-
mapping 29
```

Syntax: ...**dscp-mapping** <dscp-value>

NOTE: For complete syntax information, see “Extended ACL Syntax” on page 6-20.

Using an IP ACL to Mark ToS Values

The **dscp-marking** option for extended ACLs allows you to configure an ACL that marks matching packets with a specified ToS value.

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 1/1. Consequently, all inbound packets on interface 1/1 are marked with the specified DSCP value.

```
BigIron(config)# access-list 120 permit ip any any dscp-marking 5
BigIron(config)# interface 1/1
BigIron(config-if-1/1)# ip access-group 120 in
```

Syntax: ...**dscp-marking** <dscp-value> **802.1p-priority-marking** <0 – 7> **internal-priority-marking** <0 – 7>

The **dscp-marking** <dscp-value> parameter maps a DSCP value to an internal forwarding priority. The DSCP value can be from 0 – 63.

The **802.1p-priority-marking** <0 – 7> and **internal-priority-marking** <0 – 7> parameters apply only to the FES X-Series devices. For information about these parameters, see “Using an ACL to Change the Forwarding Queue” on page 6-45.

Using an IP ACL to Map the DSCP Value

The **dscp-cos-mapping** option applies to the FES X-Series only. This parameter 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** parameter overrides port-based priority settings.

Dropping All Fragments That Exactly Match a Flow-Based ACL

For a packet fragment that is sent to the CPU for processing, the device compares the fragment's source and destination IP addresses against the interface's ACL entries. By default, if the fragment's source and destination IP addresses exactly match an ACL entry that also has Layer 4 information (source and destination TCP or UDP application ports), the device permits or denies the fragment according to the ACL.

On an individual interface basis, you can configure an IronCore device to automatically drop a fragment whose source and destination IP addresses exactly match an ACL entry that has Layer 4 information, even if that ACL entry's action is permit. To do so, enter the following command at the configuration level for an interface:

```
BigIron(config-if-1/1)# ip access-group frag deny
```

Syntax: [no] ip access-group frag deny

NOTE: This command was added in Enterprise software release 07.5.04.

NOTE: JetCore devices also support the **ip access-group frag deny** command but the command performs a different service on JetCore devices. See "Enabling ACL Filtering of Fragmented Packets" on page 6-51.

Enabling ACL Duplication Check on Terathon Devices

NOTE: This feature is available in release 02.1.00 for the BigIron MG8 and NetIron 40G and release 02.0.02 for the NetIron IMR 640.

In releases 02.1.00 for BigIron MG8 and NetIron 40G, and in release 02.0.02 for the NetIron IMR 640, the software does not check for duplicate ACL entries. This is so the device can support the increased maximum number of ACLs. In a system with several thousand ACL entries, checking for duplicate ACL entries may consume a significant amount of time.

If desired, you can enable software checking for duplicate ACL entries. To do so, enter the following command at the Global CONFIG level of the CLI:

```
BigIron MG8(config)# acl-duplication-check
```

Syntax: [no]acl-duplication-check

ACL Accounting for the NetIron IMR 640

Multi-Service devices monitor the number of times an ACL is used to filter incoming or outgoing traffic on an interface. The **show access-list accounting** command displays the number of "hits" or how many times ACL filters permitted or denied packets that matched the conditions of the filters.

NOTE: ACL accounting does not tabulate nor display the number of Implicit denials by an ACL.

Counters, stored in hardware, keep track of the number of times an ACL filter is used.

The counters that are displayed on the ACL accounting report are:

- 1s – Number of hits during the last second. This counter is updated every second.

- 1m – Number of hits during the last minute. This counter is updated every one minute.
- 5m – Number of hits during the last five minutes. This counter is updated every five minutes.
- ac – Accumulated total number of hits. This counter begins when an ACL is bound to an interface and is updated every one minute. This total is updated until it is cleared.

  The accumulated total is updated every minute. For example, a minute after an ACL is bound to a port, it receives 10 hits per second and continues to receive 10 hits per second. After one minute, the accumulated total hits is 600. After 10 minutes, there will be 6000 hits.

The counters can be cleared when the device is rebooted, when an ACL is bound to or unbound from an interface, or by entering a **clear access-list** command.

Enabling Accounting Statistics for All ACLs

Unlike other releases of Multi-Service IronWare, in release 02.0.02 ACL accounting is not automatically enabled. Before you can collect ACL accounting statistics, you must enter the following command:

BigIron `(config)# enable-acl-counter`

Syntax: [no] enable-acl-counter

Displaying Accounting Statistics for All ACLs

To display a summary of the number of hits in all ACLs on a Multi-Service device, enter the following command:

```
NI IMR640 Router(config)#show access-list accounting brief
Collecting ACL accounting summary for VE 1 ...  Completed successfully.

ACL Accounting Summary: (ac = accumulated since accounting started)
    Int    In ACL              Total In Hit   Out ACL             Total Out Hit
    VE 1    111                473963(1s)
                               25540391(1m)
                               87014178(5m)
                               112554569(ac)
```

The display shows the following information:

This Field...	Displays...
The IP multicast traffic snooping state	The first line of the display indicates whether IP multicast traffic snooping is enabled or disabled. If enabled, it indicates if the feature is configured as passive or active.
Collecting ACL accounting summary for <interface>	Shows for which interfaces the ACL accounting information was collected and whether or not the collection was successful.
Int	The ID of the interface for which the statistics are being reported.
In ACL	The ID of the ACL used to filter the incoming traffic on the interface.
Total In Hit*	The number of hits from incoming traffic processed by all ACL entries (filters) in the ACL. A number is shown for each counter.
Out ACL	ID of the ACL used to filter the outgoing traffic on the interface.
Total Out Hit*	The number of hits from incoming traffic processed by all ACL entries (filters) in the ACL. A number is shown for each counter.

This Field...	Displays...
* The Total In Hit and Total Out Hit displays the total number of hits for all the ACL entries (or filters) in an ACL. For example, if an ACL has five entries and each entry processed matching conditions three times during the last minute, then the total Hits for the 1m counter is 15.	

Syntax: show access-list accounting brief [l2 | policy-based-routing | rate-limit]

The **l2** parameter limits the display to Layer 2 ACL accounting information.

The **policy-based-routing** parameter limits the display to policy based routing accounting information.

The **rate-limit** parameter limits the display to rate limiting ACL accounting information.

IPv4 ACL accounting statistics are displayed if no option is specified.

Displaying Statistics for an Interface

To display statistics for an interface, enter commands such as the following:

```
NI IMR640 Router(config)#show access-list accounting ve 1 in

Collecting ACL accounting for VE 1  ...  Completed successfully.

ACL Accounting Information:

Inbound: ACL 111
    1: deny tcp any any
       Hit count: (1 sec)              237000   (1 min)12502822
                  (5 min)            87014178   (accum) 99517000
    3: permit ip any any
       Hit count: (1 sec)              236961   (1 min) 13037569
                  (5 min)                   0   (accum) 13037569
    0: deny tcp 1.1.1.0 0.0.0.255 2.2.2.0 0.0.0.255
       Hit count: (1 sec)                   0   (1 min) 0
                  (5 min)                   0   (accum) 0
    2: deny udp any any
       Hit count: (1 sec)                   0   (1 min) 0
                  (5 min)                   0   (accum) 0
```

The display shows the following information:

This Field...	Displays...
The IP multicast traffic snooping state	The first line of the display indicates whether IP multicast traffic snooping is enabled or disabled. If enabled, it indicates if the feature is configured as passive or active.
Collecting ACL accounting summary for <interface>	Shows the interface included in the report and whether or not the collection was successful.
Outbound/Inbound ACL ID	Shows the direction of the traffic on the interface and the ID of the ACL used.

This Field...	Displays...
#	Shows the index of the ACL entry, starting with 0, followed by the permit or deny condition defined for that ACL entry. (The first entry created for an ACL is assigned the index 0. The next one created is indexed as 1, and so on.) ACL entries are arranged beginning with the entry with the highest number of hits for IPv4 ACLs. For all other options, ACL entries are displayed in order of ascending ACL filter IDs.
Hit count	Shows the number of hits for each counter.

Syntax: show access-list accounting ethernet [<slot>/<port> | ve <ve-number>] in | out [l2 | policy-based-routing | rate-limit]

Use **ethernet** <slot>/<port> to display a report for a physical interface.

Use **ve** <ve-number> to display a report for the ports that are included in a virtual routing interface. For example, if ports 1/2, 1/4, and 1/6 are all members of ve 2, the report includes information for all three ports.

Use the **in** parameter to display statistics for incoming traffic; **out** for outgoing traffic.

The **l2** parameter limits the display to Layer 2 ACL accounting information.

The **policy-based-routing** parameter limits the display to policy based routing accounting information. This option is only available for incoming traffic.

The **rate-limit** parameter limits the display to rate limiting ACL accounting information.

Clearing the ACL Statistics

Statistics on the ACL account report can be cleared:

- When a software reload occurs
- When the ACL is bound to or unbound from an interface
- When you enter the **clear access-list** command, as in the following example:

```
NI IMR640 Router(config)# clear access-list all
```

Syntax: clear access-list all | ethernet <slot>/<port> | ve <ve-num>

Enter **all** to clear all statistics for all ACLs.

Use **ethernet** <slot>/<port> to clear statistics for ACLs a physical port.

Use **ve** <ve-number> to clear statistics for all ACLs bound to ports that are members of a virtual routing interface.

Enabling ACL Filtering of Fragmented Packets

Filtering Fragmented Packets for Rule-Based ACLs (JetCore)

By default, when a rule-based ACL is applied to a port, the port will use the ACL to permit or deny the first fragment of a fragmented packet. Generally, denying the first fragment of a packet is sufficient, since a transaction cannot be completed without the entire packet. However, the action applied on the first packet may not be the same action applied on subsequent fragments of the same packet. For example, if the first fragmented packet is permitted, the next fragment of the packet may be wrongfully denied.

For tighter control, you can enable CPU filtering of all packet fragments on a port. When you enable CPU filtering, the port sends all the fragments of a fragmented packet to the CPU. The CPU then permits or denies each fragment according to the ACL applied to the port. You can enable CPU filtering of fragments on individual ports.

You also can configure the port to drop all packet fragments.

NOTE: The fragmentation support described in this section applies only to JetCore devices and only to rule-based ACLs.

NOTE: Enhanced fragment handling is not supported on 10 Gigabit Ethernet modules. By default, 10 Gigabit Ethernet modules also forward the first fragment instead of using the ACLs to permit or deny the fragment.

To enable CPU filtering of packet fragments on an individual port, enter commands such as the following:

```
BigIron(config)# interface ethernet 1/1
BigIron(config-if-1/1)# ip access-group frag inspect
```

Syntax: [no] ip access-group frag inspect | deny

The **inspect | deny** parameter specifies whether you want fragments to be sent to the CPU or dropped:

- **inspect** – This option sends all fragments to the CPU.
- **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.

NOTE: IronCore devices also support the **ip access-group frag deny** command but the command performs a different service on IronCore devices. See “Dropping All Fragments That Exactly Match a Flow-Based ACL”.

Throttling the Fragment Rate

By default, when you enable CPU filtering of packet fragments on a JetCore device, all fragments are sent to the CPU. Normally, the fragment rate in a typical network does not place enough additional load on the CPU to adversely affect performance. However, performance can be affected if the device receives a very high rate of fragments. For example, a misconfigured server or a hacker can affect the device’s performance by flooding the CPU with fragments.

You can protect against fragment flooding by specifying the maximum number of fragments the device or an individual interface is allowed to send to the CPU in a one-second interval. If the device or an interface 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. In addition, the device starts a holddown timer and continues to either drop or forward fragments until the holddown time expires.

The device also generates a Syslog message.

To specify the maximum fragment rate per second, enter commands such as the following:

```
BigIron(config)# ip access-list frag-rate-on-system 15000 exceed-action drop
reset-interval 10
BigIron(config)# ip access-list frag-rate-on-interface 5000 exceed-action forward
reset-interval 5
```

The first command sets the fragment threshold at 15,000 per second, for the entire device. If the device receives more than 15,000 packet fragments in a one-second interval, the device takes the specified action. The action specified with this command is to drop the excess fragments and continue dropping fragments for a holddown time of ten minutes. After the ten minutes have passed, the device starts sending fragments to the CPU again for processing.

The second command sets the fragment threshold at 5,000 for individual interfaces. If any interface on the device receives more than 5,000 fragments in a one-second interval, the device takes the specified action. In this case, the action is to forward the fragments in hardware without filtering them. The device continues forwarding fragments in hardware for five minutes before beginning to send fragments to the CPU again.

Both thresholds apply to the entire device. Thus, if an individual interface's fragment threshold is exceeded, the drop or forward action and the holddown time apply to all fragments received by the device.

Syntax: [no] ip access-list frag-rate-on-system <num> exceed-action drop | forward reset-interval <mins>

and

Syntax: [no] ip access-list frag-rate-on-interface <num> exceed-action drop | forward reset-interval <mins>

The <num> parameter specifies the maximum number of fragments the device or an individual interface can receive and send to the CPU in a one-second interval.

- **frag-rate-on-system** – Sets the threshold for the entire device. The device can send to the CPU only the number of fragments you specify per second, regardless of which interfaces the fragments come in on. If the threshold is exceeded, the device takes the exceed action you specify.
- **frag-rate-on-interface** – Sets the threshold for individual interfaces. If an individual interface receives more than the specified maximum number of fragments, the device takes the exceed action you specify.

The <num> parameter specifies the maximum number of fragments per second.

- For **frag-rate-on-system**, you can specify from 600 – 12800. The default is 6400.
- For **frag-rate-on-interface**, you can specify from 300 – 8000. The default is 4000.

The **drop | forward** parameter specifies the action to take if the threshold (<num> parameter) is exceeded:

- **drop** – fragments are dropped without filtering by the ACLs
- **forward** – fragments are forwarded in hardware without filtering by the ACLs

The <mins> parameter specifies the number of minutes the device will enforce the drop or forward action after a threshold has been exceeded. You can specify from 1 – 30 minutes, for **frag-rate-on-system** or **frag-rate-on-interface**.

Syslog Messages for Exceeded Fragment Thresholds

If a fragment threshold is exceeded, the device generates one of the following Syslog messages.

Table 6.6: Syslog Messages for Exceeded Fragment Threshold

Message Level	Message	Explanation
Notification	ACL system fragment packet inspect rate <rate> exceeded	The <rate> indicates the maximum rate allowed.
Notification	ACL port fragment packet inspect rate <rate> exceeded on port <portnum>	The <rate> indicates the maximum rate allowed. The <portnum> indicates the port.

Filtering Fragmented or Non-Fragmented Packets on the NetIron IMR 640

NOTE: This topic applies to NetIron IMR 640 running releases 02.0.02 and later.

Multi-Service devices can filter fragmented and non-fragmented packets using extended ACLs.

To define an extended ACL to deny or permit traffic with fragmented or unfragmented packets, enter a command such as those shown in one of the methods below:

Numbered ACLs

```
NI IMR640 Router(config)# access-list 111 deny ip any any fragment

NI IMR640 Router(config)# int eth 1/1
NI IMR640 Router(config-if-1/1)# ip access-group 111 in
```



```
NI IMR640 Router(config)# write memory
```

The first line in the example defines ACL 111 to deny any fragmented packets. Other packets will be denied or permitted, based on the next filter condition.

Next, after assigning the ACL to Access Group 111, the access group is bound to port 1/1. It will be used to filter incoming traffic.

Syntax: access-list <acl-num> deny | permit <ip-protocol> <source-ip> | <hostname> <wildcard> [<operator> <source-tcp/udp-port>] <destination-ip> | <hostname> [<icmp-type> | <num>] <wildcard> [<operator> <destination-tcp/udp-port>] [precedence <name> | <num>] [tos <name> | <num>] [ip-pkt-len <value>] [log] [fragment] | [non-fragmented]

The <acl-num> parameter identifies the numbered or the ACL. Enter a number from 100 - 199 for extended ACLs.

Enter the **fragment** parameter to allow the ACL to filter fragmented packets. Use the **non-fragmented** parameter to filter non-fragmented packets.

NOTE: The **fragmented** and **non-fragmented** parameters cannot be used together in an ACL entry.

Complete the configuration by specifying options for the ACL entry. Options you can use are discussed in .....

Named ACLs

```
NI IMR640 Router(config)# ip access-list extended melon deny ip any any fragment
```

```
NI IMR640 Router(config)# int eth 1/1
NI IMR640 Router(config-if-1/1)# ip access-group melon in
NI IMR640 Router(config)# write memory
```

The first line in the example defines ACL melon to deny any fragmented packets. Other packets will be denied or permitted, based on the next filter condition.

Next, after assigning the ACL to Access Group melon, the access group is bound to port 1/1. It will be used to filter incoming traffic.

Syntax: ip access-list extended <acl-name> | <acl-num> deny | permit <ip-protocol> <source-ip> | <hostname> <wildcard> [<operator> <source-tcp/udp-port>] <destination-ip> | <hostname> [<icmp-type> | <num>] <wildcard> [<operator> <destination-tcp/udp-port>] [precedence <name> | <num>] [tos <name> | <num>] [ip-pkt-len <value>] [log] [fragment] | [non-fragmented]

Enter **extended** to indicate the named ACL is an extended ACL.

The <acl-name> | <acl-num> parameter allows you to specify an ACL name or number. If using a name, 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, enter a number from 100 – 199 for extended ACLs.

Enter the **fragment** parameter to allow the ACL to filter fragmented packets. Use the **non-fragmented** parameter to filter non-fragmented packets.

NOTE: The **fragmented** and **non-fragmented** parameters cannot be used together in an ACL entry.

Complete the configuration by specifying options for the ACL entry. Options you can use are discussed in the *Foundry Enterprise Configuration and Management Guide*.

Enabling Hardware Filtering for Packets Denied by Flow-Based ACLs

By default, packets denied by ACLs are filtered by the CPU. You can enable the device to create CAM entries for packets denied by ACLs. This causes the filtering to occur in hardware instead of in the CPU.

When you enable hardware filtering of denied packets, the first time the device filters a packet denied by an ACL, the device sends the packet to the CPU for processing. The CPU also creates a CAM entry for the denied packet.

Subsequent packets with the same address information are filtered using the CAM entry. The CAM entry ages out after two minutes if not used.

To enable hardware filtering of denied packets, enter the following command at the global CONFIG level of the CLI:

```
BigIron(config)# hw-drop-acl-denied-packet
```

Syntax: [no] hw-drop-acl-denied-packet

Enabling Strict TCP or UDP Mode for Flow-Based ACLs

By default, when you use ACLs to filter TCP or UDP traffic, the Foundry device does not compare all TCP or UDP packets against the ACLs.

For TCP and UDP, the device first compares the source and destination information in a TCP control packet or a UDP packet against entries in the session table. The session table contains forwarding entries based on Layer 3 and Layer 4 information.

- If the session table contains a matching entry, the device forwards the packet, assuming that the first packet the device received with the same address information was permitted by the ACLs.
- If the session table does not contain a matching entry, the device sends the packet to the CPU, where the software compares the packet against the ACLs. If the ACLs permit the packet (explicitly by a permit ACL entry or implicitly by the absence of a deny ACL entry), the CPU creates a session table entry for the packet's forwarding information and forwards the packet.

For TCP, this behavior by default applies only to control packets, not to data packets. Control packets include packet types such as SYN (Synchronization) packets, FIN (Finish) packets, and RST (Reset) packets.

For tighter access or forwarding control, you can enable the device to perform strict TCP or UDP ACL processing. The following sections describe the strict modes in more detail.

Enabling Strict TCP Mode

By default, when you use ACLs to filter TCP traffic, the Foundry device does not compare all TCP packets against the ACLs. Instead, the device compares TCP control packets against the ACLs, but not data packets. Control packets include packet types such as SYN (Synchronization) packets, FIN (Finish) packets, and RST (Reset) packets.

In normal TCP operation, TCP data packets are present only if a TCP control session for the packets also is established. For example, data packets for a session never occur if the TCP SYN for that session is dropped. Therefore, by filtering the control packets, the Foundry device also implicitly filters the data packets associated with the control packets. This mode of filtering optimizes forwarding performance for TCP traffic by forwarding data packets without examining them. Since the data packets are present in normal TCP traffic only if a corresponding TCP control session is established, comparing the packets for the control session to the ACLs is sufficient for filtering the entire session including the data.

However, it is possible to generate TCP data packets without corresponding control packets, in test or research situations for example. In this case, the default ACL mode does not filter the data packets, since there is no corresponding control session to filter. To filter this type of TCP traffic, use the strict ACL TCP mode. This mode compares all TCP packets to the configured ACLs, regardless of whether the packets are control packets or data packets. If the ACLs permit the packet, the device creates a session entry for forwarding other TCP packets with the same Layer 3 and Layer 4 addresses.

NOTE: Regardless of whether the strict mode is enabled or disabled, the device always compares TCP control packets against the configured ACLs before creating a session entry for forwarding the traffic.

NOTE: If the device's configuration currently has ACLs associated with interfaces, remove the ACLs from the interfaces before changing the ACL mode.

To enable the strict ACL TCP mode, enter the following command at the global CONFIG level of the CLI:


```
BigIron(config)# ip strict-acl-tcp
```

Syntax: [no] ip strict-acl-tcp

This command configures the device to compare all TCP packets against the configured ACLs before forwarding them.

To disable the strict ACL mode and return to the default ACL behavior, enter the following command:

```
BigIron(config)# no ip strict-acl-tcp
```

NOTE: If you are using software release 07.6.01 or later, enter the **ip rebind-acl** command at the global CONFIG level of the CLI to place the **ip strict-acl-tcp** or **no ip strict-acl-tcp** command into effect.

Enabling Strict UDP Mode

By default, when you use ACLs to filter UDP traffic, the Foundry device does not compare all UDP packets against the ACLs. Instead, the device compares the source and destination information against entries in the session table. The session table contains forwarding entries based on Layer 3 and Layer 4 information.

- If the session table contains a matching entry, the device forwards the packet, assuming that the first packet the device received that contains the same address information was permitted by the ACLs.
- If the session table does not contain a matching entry, the device sends the packet to the CPU, where the software compares the packet against the ACLs. If the ACLs permit the packet (explicitly by a permit ACL entry or implicitly by the absence of a deny ACL entry), the CPU creates a session table entry for the packet's forwarding information and forwards the packet.

For tighter control, the software provides the strict ACL UDP mode. When you enable strict UDP processing, the device sends every UDP packet to the CPU and compares the packet against the configured ACLs.

NOTE: If the device's configuration currently has ACLs associated with interfaces, remove the ACLs from the interfaces before changing the ACL mode.

To enable the strict ACL UDP mode, enter the following command at the global CONFIG level of the CLI:

```
BigIron(config)# ip strict-acl-udp
```

Syntax: [no] ip strict-acl-udp

This command configures the device to compare all UDP packets against the configured ACLs before forwarding them.

To disable the strict ACL mode and return to the default ACL behavior, enter the following command:

```
BigIron(config)# no ip strict-acl-udp
```

NOTE: If you are using software release 07.6.01 or later, enter the **ip rebind-acl** command at the global CONFIG level of the CLI to place the **ip strict-acl-udp** or **no ip strict-acl-udp** command into effect.

Configuring ACL Packet and Flow Counters

You can configure counters for packets and flows that match entries in an ACL. Using the CLI, you can display the contents of the counters and clear them.

- The ACL packet counter feature provides an accurate count of packets matching individual ACL entries.
- The ACL flow counter feature provides an approximate count of flows matching individual ACL entries. This feature can be used for troubleshooting purposes to provide an indication of flow activity against an ACL. Each time the Foundry device receives the first packet of a flow matching an entry in an ACL list, the flow counter for that ACL entry is incremented by one. If a flow lasts longer than two minutes, the flow counter for the ACL entry is incremented again.

NOTE: The ACL flow counter feature is designed to monitor the general volume of flow activity for an ACL. It is not intended to be used for accounting purposes.

The ACL flow and packet counters are incremented differently depending on whether packets are handled by the Management Processor (MP) or the POS processor, and whether they are permit or deny flows.

The Management Processor (MP) handles flows as follows:

For flows handled by the Management Processor:

- For permit flows, only flows are counted. If a permit flow lasts longer than two minutes, the flow counter is incremented again.
- For deny flows, only packets are counted.

For flows handled by the POS processor (flows passing through POS ports):

- For permit flows, both flows and packets are counted. If a permit flow lasts longer than two minutes, the flow counter is incremented again.
- For deny flows, only packets are counted.

By default the ACL packet and flow counters are disabled. To activate them, enter the following command:

```
BigIron(config)# enable-acl-counter
```

Syntax: [no] enable-acl-counter

Once the ACL packet and flow counters are enabled, you can disable them with the **no** form of the **enable-acl-counter** command. Disabling and then re-enabling the ACL packet and flow counters resets them to zero.

To display the packet and flow counters for ACL 100:

```
BigIron# show access-list 100
Extended IP access list 100 (Total flows: 432, Total packets: 42000)
  permit tcp 1.1.1.0 0.0.0.255 any (Flows: 80, Packets: 12900)
  deny udp 1.1.1.0 0.0.0.255 any (Flows: 121, Packets: 20100)
  permit ip 2.2.2.0 0.0.0.255 any (Flows: 231, Packets: 9000)
```

Syntax: show access-list <acl-num> | <acl-name> | all

To clear the flow counters for ACL 100:

```
BigIron# clear access-list 100
```

Syntax: clear access-list <acl-num> | <acl-name> | all

NOTE: When an ACL is modified, the ACL flow counters sent by a POS module to the management module are ignored for one minute. This allows the POS module and the management module time to synchronize after the ACL is modified.

Filtering on IP Precedence and ToS Values of Flow-Based ACLs

You can configure an extended IP ACL to filter traffic based on IP precedence. Enter commands such as the following:

```
BigIron(config)# access-list 103 deny tcp 209.157.21.0/24 209.157.22.0/24
precedence internet
BigIron(config)# access-list 103 deny tcp 209.157.21.0/24 eq ftp 209.157.22.0/24
precedence 6
BigIron(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:

```
BigIron(config)# access-list 104 deny tcp 209.157.21.0/24 209.157.22.0/24 tos normal
BigIron(config)# access-list 104 deny tcp 209.157.21.0/24 eq ftp 209.157.22.0/24 tos 13
BigIron(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.

ACL Filtering for Traffic Switched Within a Virtual Routing Interface

By default, a Foundry device does not filter traffic that is switched from one port to another within the same virtual routing interface, even if a flow-based or rule-based ACL is applied to the interface. You can enable the device to filter switched traffic within a virtual routing interface. When you enable the filtering, 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. This feature does not apply to ACLs applied to outbound traffic.

To enable filtering of traffic switched within a virtual routing interface, enter the following command at the configuration level for the interface:

```
BigIron(config-vif-1)# ip access-group ve-traffic
```

Syntax: [no] ip access-group ve-traffic

NOTE: The **ve-traffic** command is applied to a physical port. If you configure this command on a virtual routing interface that is a member of a tagged VLAN and ACLs are applied to that tagged VLAN, the **ve-traffic** command will not see the ACLs. The traffic will not be filtered. Ensure that the virtual routing interface is does not belong to a tagged VLAN.

Using Flow-Based ACLs to Filter ARP Packets

Starting with software release 07.6.034, you can use ACLs to filter ARP packets. Without this feature, ACLs cannot be used to permit or deny incoming or outgoing ARP packets. (Although an ARP packet contains an IP address just as an IP packet does, it is not an IP packets and is not subject to the normal filtering provided by ACLs.)

When a Foundry device receives an ARP request, the source MAC and IP addresses are stored in the device's ARP table. A new record in the ARP table overwrites existing records that contain the same IP address. This behavior can cause a condition called "ARP hijacking", when two hosts with the same IP address try to send an ARP request to the Foundry device.

Normally ARP hijacking is not a problem because IP assignments are done dynamically; however, in some cases, such as when the **ip follow** command is used, ARP hijacking can occur.

The **ip follow** command allows a router interface to share the IP address of another router interface. **Ip follow** conserves IP addresses, while separating Layer 2 traffic from different sources by port-based VLAN. Since multiple VLANs and the router interfaces that are associated with each of the VLANs share the same IP segment, it is possible for two hosts in two different VLANs to fight for the same IP address in that segment. ARP filtering using ACLs protects an IP host's record in ARP table from being overwritten by a hijacking host.

Using ACLs to filter ARP request checks the source IP address in the received ARP packet. Only packets with the permitted IP address will be allowed to be to be written in the ARP table; others are dropped.

Configuration Considerations:

- This feature is available on all devices running Layer 3 code. On a VM1 module, this filtering occurs on the management processor.
- The feature is available on physical interfaces and virtual routing interfaces. It is supported on the following physical interface types: Ethernet, POS, ATM, and trunks.
- ACLs used to filter ARP packets a virtual routing interface can be inherited from a previous interface if the virtual routing interface is defined as a follower virtual routing interface

Configuring ACLs for ARP Filtering

To implement the ACL ARP filtering feature, enter commands such as the following:

```
BigIron(config)# access-list 101 permit ip host 192.168.2.2 any
BigIron(config)# access-list 102 permit ip host 192.168.2.3 any
BigIron(config)# access-list 103 permit ip host 192.168.2.4 any
BigIron(config)# vlan 2
BigIron(config-vlan-2)# tag ethe 1/1 to 1/2
BigIron(config-vlan-2)# router-interface ve 2
BigIron(config-vlan-2)# vlan 3
BigIron(config-vlan-3)# tag ethe 1/1 to 1/2
BigIron(config-vlan-3)#router-int ve 3
BigIron(config-vlan-3)# vlan 4
BigIron(config-vlan-4)# tag ethe 1/1 to 1/2
BigIron(config-vlan-4)# router-int ve 4
BigIron(config-vlan-4)# interface ve 2
BigIron(config-ve-2)# ip access-group 101 in
BigIron(config-ve-2)# ip address 192.168.2.1/24
BigIron(config-ve-2)# ip use-acl-on-arp 103
BigIron(config-ve-2)# exit
BigIron(config)# interface ve 3
BigIron(config-ve-3)# ip access-group 102 in
BigIron(config-ve-3)# ip follow ve 2
BigIron(config-ve-3)# no ip follow acl
BigIron(config-ve-3)# ip use-acl-on-arp
BigIron(config-ve-3)# exit

BigIron(config-vlan-4)# interface ve 4
BigIron(config-ve-4)# ip follow ve 2
BigIron(config-ve-4)# ip use-acl-on-arp
BigIron(config-ve-4)# exit
```

Syntax: [no] ip use-acl-on-arp [<access-list-number>]

When the **use-acl-on-arp** command is configured, the ARP module checks the source IP address of the ARP request packets received on the interface. It then applies the specified ACL policies to the packet. Only the packet with the IP address that the ACL permits will be allowed to be to be written in the ARP table; those that are not permitted will be dropped.

The <access-list-number> parameter identifies the ID of the standard ACL that will be used to filter the packet. Only the source and destination IP addresses will be used to filter ARP packet. You can do one of the following for <access-list-number>:

- Enter an ACL ID to explicitly specify the ACL to be used for filtering. In the example above, the line BigIron`(config-ve-2)# ip use-acl-on-arp 103` specifies ACL 103 to be used as the filter.
- Allow the ACL ID to be inherited from the IP ACLs that have been defined for the device. In the example above, the line BigIron`(config-ve-3)# ip use-acl-on-arp` does not define an ACL, but allows the ACL to be inherited from the IP ACL 102. Also in the example, the line BigIron`(config-ve-4)# ip use-acl-on-arp` allows the ACL to be inherited from IP ACL 101 because of the ip follow relationship between virtual routing interface 2 and virtual routing interface 4. Virtual routing interface 2 is configured with IP IP ACL 101; thus virtual routing interface 4 inherits IP ACL 101.

ARP requests will not be filtered by ACLs if one of the following conditions occur:

- If the ACL is to be inherited from an IP ACL, but there is no IP ACL defined.
- An ACL ID is specified for the **use-acl-on-arp** command, but no IP address or “any any” filtering criteria have been defined under the ACL ID.

Displaying ACL Filters for ARP

To determine what ACLs have been configured to filter ARP requests, enter a command such as the following:

```
BigIron(config)# show acl-on-arp
Port  ACL ID  Filter Count
2     103     10
3     102     23
4     101     12
```

Syntax: show acl-on-arp [ethernet [<portnum>] | loopback [<num>] | ve [<num>]]

If port number or the interface number is not specified, all ports on the device that use ACLs for ARP filtering will be included in the display.

The Filter Count column shows how many ARP packets have been dropped on the interface since the last time the count was cleared.

Clearing ARP Filter Count

To clear the filter count for all interfaces on the device, enter a command such as the following:

```
BigIron(config)# clear acl-on-arp
```

Syntax: clear acl-on-arp

The command resets the filter count on all interfaces in a device back to zero

ACLs and ICMP

ACLs can be used to filter traffic based on ICMP packets. This section presents the following sections related to ICMP:

Using Flow-Based ACLs to Filter ICMP Packets Based on the IP Packet Length

NOTE: This feature is supported in software releases 07.7.00 and later.

To configure an extended ACL that filters based on the IP packet length of ICMP packets, enter commands such as the following:

```
BigIron(config)# access-list 105 deny icmp echo any any ip-pkt-len 92
BigIron(config)# access-list 105 deny icmp echo any any ip-pkt-len 100
BigIron(config)# access-list 105 permit ip any any
```

The commands in this example deny (drop) ICMP echo request packets that contain a total length of 92 or 100 in the IP header field. You can specify an IP packet length of 1 – 65535. See the section “ICMP Filtering with Flow-Based ACLs” on page 6-61 for additional information on using ICMP to filter packets.

ICMP Filtering with Flow-Based ACLs

Most Foundry software releases that support flow-based ACLs filter traffic based on the following ICMP message types:

- echo
- echo-reply
- information-request

- mask-reply
- mask-request
- parameter-problem
- redirect
- source-quench
- time-exceeded
- timestamp-reply
- timestamp-request
- unreachable
- <num>

In Enterprise IronWare softeware release 07.8.00 and later, additional ICMP message types have been added to the CLI. Also, to create ACL policies that filter ICMP message types, you can either enter the description of the message type or enter its type and code IDs. Furthermore ICMP message type filtering is now available for rule-based ACLs on BigIron Layer 2 Switch and Layer 3 Switch images.

Numbered ACLs

For example, to deny the echo message type in a numbered ACL, enter commands such as the following when configuring a numbered ACL:

```
BigIron(config)# access-list 109 deny ICMP any any echo
```

or

```
BigIron(config)# access-list 109 deny ICMP any any 8 0
```

Syntax: [no] access-list <num>

Syntax: deny | permit icmp
<source-ip-address> | <source-ip-address/subnet-mask> | any | host <source-host>
<destination-ip-address> | destination-ip-address/subnet-mask> | any | host <destination-host>
<icmp-type> | <icmp-type-number> <icmp-code-number>

The **deny | permit** parameter indicates whether packets that match the policy are dropped or forwarded.

You can either enter the name of the message type for <icmp-type> or the type number and code number of the message type. Refer to Table 6.7 on page 6-63 for valid values.

Named ACLs

For example, to deny the administratively-prohibited message type in a named ACL, enter commands such as the following:

```
BigIron(config)# ip access-list extended melon
BigIron(config-ext-nacl)# deny ICMP any any administratively-prohibited
```

or

```
BigIron(config)# ip access-list extended melon
BigIron(config-ext-nacl)#deny ICMP any any 3 13
```

Syntax: [no] ip access-list extended <acl-num> | <acl-name>

Syntax: deny | permit icmp
<source-ip-address> | <source-ip-address/subnet-mask> | any | host <source-host>
<destination-ip-address> | destination-ip-address/subnet-mask> | any | host <destination-host>
<icmp-type> | <icmp-type-number> <icmp-code-number>

The **extended** parameter indicates the ACL entry is an extended ACL.

The <acl-name> | <acl-num> parameter allows you to specify an ACL name or number. If using a name, 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, enter a number from 100 – 199 for extended ACLs.

The **deny | permit** parameter indicates whether packets that match the policy are dropped or forwarded.

You can either use the <icmp-type> and enter the name of the message type or use the the <icmp-type-number> <icmp-ode-number> parameter and enter the type number and code number of the message. Refer to Table 6.7 for valid values.

NOTE: “X” in the Type-Number or Code-Number column in Table 6.7 means the device filters any traffic of that ICMP message type.

Table 6.7: ICMP Message Types and Codes

ICMP Message Type	Type	Code
administratively-prohibited	3	13
any-icmp-type	x	x
destination-host-prohibited	3	10
destination-host-unknown	3	7
destination-net-prohibited	3	9
destination-network-unknown	3	6
echo	8	0
echo-reply	0	0
general-parameter-problem **Note:** This message type indicates that required option is missing.	12	1
host-precedence-violation	3	14
host-redirect	5	1
host-tos-redirect	5	3
host-tos-unreachable	3	12
host-unreachable	3	1
information-request	15	0
log		
mask-reply	18	0
mask-request	17	0
net-redirect	5	0
net-tos-redirect	5	2
net-tos-unreachable	3	11

Table 6.7: ICMP Message Types and Codes

ICMP Message Type	Type	Code
net-unreachable	3	0
packet-too-big	3	4
parameter-problem **Note:** This message includes all parameter problems	12	0
port-unreachable	3	3
precedence-cutoff	3	15
protocol-unreachable	3	2
reassembly-timeout	11	1
redirect **Note:** This includes all redirects.	5	x
router-advertisement	9	0
router-solicitation	10	0
source-host-isolated	3	8
source-quench	4	0
source-route-failed	3	5
time-exceeded	11	x
timestamp-reply	14	0
timestamp-request	13	0
ttl-exceeded	11	0
unreachable **Note:** This includes all unreachable messages	3	x

Enabling ICMP Unreachable Messages for Traffic Denied by Flow-Based ACLs

By default, a Foundry device does not send a message to another device when an ACL on the Foundry device denies a packet from the other device. You can enable a Layer 3 Switch to send an ICMP unreachable message to a device when an ACL denies a packet from the device.

To enable the ICMP unreachable messages, enter the following command at the global CONFIG level of the CLI:

```
BigIron(config)# acl-denied-icmp-msg
```

Syntax: [no] acl-denied-icmp-msg

The command applies globally to all ACLs configured on the device.

NOTE: This command applies only to Layer 3 Switches.

NOTE: This command does not take effect in the following cases:

—Rule-based ACLs are enabled.

—The **hw-drop-acl-denied-packet** command is in effect.

In either case, all packets denied by the ACL are dropped by hardware without sending an ICMP message.

ICMP Filtering for Extended ACLs on the NetIron IMR 640

NOTE: This feature applies to release 02.0.02 for the NetIron IMR 640.

In this release, extended ACL policies can be created to filter traffic based on its ICMP message type. You can either enter the description of the message type or enter its type and code IDs. All packets matching the defined ICMP message type or type number and code number are processed in hardware.

Numbered ACLs

For example, to deny the echo message type in a numbered, extended ACL, enter commands such as the following when configuring a numbered ACL:

```
NI IMR640 Router(config)# access-list 109 deny ICMP any any echo
```

or

```
NI IMR640 Router(config)# access-list 109 deny ICMP any any 8 0
```

Syntax: [no] access-list <num> deny | permit icmp any any [log] <icmp-type> | <type-number> <code-number>

The **deny | permit** parameter indicates whether packets that match the policy are dropped or forwarded.

You can either enter the name of the message type for <icmp-type> or the message's <type number> and <code number> of the message type. See Table 6.7 on page 6-63 for valid values.

Named ACLs

For example, to deny the administratively-prohibited message type in a named ACL, enter commands such as the following:

```
BigIron(config)# ip access-list extended melon
BigIron(config-ext-nacl)# deny ICMP any any administratively-prohibited
```

or

```
BigIron(config)# ip access-list extended melon
BigIron(config-ext-nacl)#deny ICMP any any 3 13
```

Syntax: [no] ip access-list extended <acl-name>
deny | permit host icmp any any [log] <icmp-type> | <type-number> <code-number>

The **extended** parameter indicates the ACL entry is an extended ACL.

The <acl-name> | <acl-num> parameter allows you to specify an ACL name or number. If using a name, 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, enter a number from 100 – 199 for extended ACLs.

The **deny | permit** parameter indicates whether packets that match the policy are dropped or forwarded.

You can either use the <icmp-type> and enter the name of the message type or use the <type-number> <code-number> parameter to enter the type number and code number of the message. See Table 6.7 on page 6-63 for valid values.

Using ACLs and NAT on the Same Interface (Flow-Based ACLs)

NOTE: These guidelines do not apply to devices that are using the VM1 management module. You can configure ACLs and NAT on the same port without having to follow these guidelines.

You can use ACLs and NAT on the same interface, as long as you follow these guidelines:

- You must use the **ip strict-acl-tcp** command when configuring ACLs and NAT is configured on the same Layer 2 Switch. (See the instructions below on how to use this command.)
- Do not enable NAT on an interface until you have applied ACLs (as described below) to the interface. If NAT is already enabled, you must disable it, apply the ACLs, then re-enable NAT on the interface.
- Enable the strict TCP mode.
- On the inside NAT interface (the one connected to the private addresses), apply inbound ACLs that permit TCP, UDP, and ICMP traffic to enter the device from the private sub-net.

You can use a standard ACL to permit all traffic (including TCP, UDP, and ICMP traffic) or an extended ACL with separate entries to explicitly permit TCP, UDP, and ICMP traffic.

NOTE: You do not need to apply ACLs to permit TCP, UDP, and ICMP traffic unless you are applying other ACLs to the interface as well. If you do not plan to apply any ACLs to a NAT interface, then you do not need to apply the ACLs to permit TCP, UDP, and ICMP traffic.

Here is an example of how to configure device to use ACLs and NAT on the same interfaces. In this example, the inside NAT interface is port 1/1 and the outside NAT interface is port 2/2.

The following commands enable the strict TCP mode and configure an ACL to permit all traffic from the 10.10.200.x sub-net. A second ACL denies traffic from a specific host on the Internet.

```
BigIron(config)# ip strict-acl-tcp
BigIron(config)# access-list 1 permit 10.10.200.0 0.0.0.255
BigIron(config)# access-list 2 deny 209.157.2.184
```

The following commands configure global NAT parameters.

```
BigIron(config)# ip nat inside source list 1 pool outadds overload
BigIron(config)# ip nat pool outadds 204.168.2.1 204.168.2.254 netmask 255.255.255.0
```

The following commands configure the inside and outside NAT interfaces. Notice that the ACLs are applied to the inbound direction on the inside NAT interface, and are applied ***before*** NAT is enabled. In this example, ACL 1 permits all traffic to come into the inside interface from the private sub-net. ACL 2 denies traffic from a specific host from going out the interface to the private sub-net.

```
BigIron(config)# interface ethernet 1/1
BigIron(config-if-1/1)# ip address 10.10.200.1 255.255.255.0
BigIron(config-if-1/1)# ip access-group 1 in
BigIron(config-if-1/1)# ip access-group 2 out
BigIron(config-if-1/1)# ip nat inside
BigIron(config-if-1/1)# interface ethernet 2/2
BigIron(config-if-2/2)# ip address 204.168.2.78 255.255.255.0
BigIron(config-if-2/2)# ip nat outside
```

NOTE: If you are using software release 07.6.01 or later, enter the **ip rebind-acl** command at the global CONFIG level of the CLI to place the **ip strict-acl-tcp** command into effect.

Troubleshooting Rule-Based ACLs

Use the following methods to troubleshoot a rule-based ACL:

- 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 the Number of Layer 4 CAM Entries” on page 6-12.
- To view the types of packets being received on an interface, enable ACL statistics using the **enable-acl-counter** command, reapply the ACLs by entering the **ip rebind-acl all** command, then display the statistics by entering the **show ip acl-traffic** command.

- To determine whether an ACL entry is correctly matching packets, add the **log** option to the ACL entry, then reapply the ACL. This forces the device to send packets that match the ACL entry to the CPU for processing. The **log** option also generates a Syslog entry for packets that are permitted or denied by the ACL entry.
- 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.

Using IP Receive Access List to Filter Packets

The *IP receive access list* feature uses IPv4 ACLs to filter the packets intended for the management process to protect the management module from being overloaded with heavy traffic that was sent to one of the Layer 3 Switch IP interfaces.

NOTE: This feature is available on the BigIron MG8 and NetIron 40G running software release 02.2.01 and later. It applies to IPv4 unicast and multicast packets.

Configuring IP Receive Access List

IP receive access list is a global configuration command. Once it is applied, the command will be effective on all the management modules on the device. To configure the feature, do the following:

1. Create a numbered ACL that will be used as the IP receive ACL. This ACL can be a standard (1– 99) or extended (100–199) ACL. Named ACLs are not supported.

   For example,

   ```
   BigIron MG8(config)# access-list 10 deny host 209.157.22.26 log
   BigIron MG8(config)# access-list 10 deny 209.157.29.12 log
   BigIron MG8(config)# access-list 10 deny host IPHost1 log
   BigIron MG8(config)# access-list 10 permit any
   BigIron MG8(config)# write memory
   ```

2. Configure ACL 10 as the IP receive access list by entering the following command:

   ```
   BigIron MG8(config)# ip receive access-list 10
   ```

   Syntax: [no] ip receive access-list <num>

   Specify an access list number for <num>.

   The IP receive ACL is applied globally to all interfaces on the device.

Displaying IP Receive Access List

To determine if IP receive access list has been configured on the device, enter the following command:

```
BigIron MG8# show access-list bindings
L4 configuration:

ip receive access-list 101
```

Syntax: show access-list bindings

Chapter 7
Hardware-Based Policy-Based Routing

NOTE: This feature is supported on JetCore devices and on the BigIron MG8 running software release 01.0.01 or later.

Hardware-based Policy-Based Routing (PBR) routes traffic in hardware based on policies you define. A PBR policy specifies the next hop for traffic that matches the policy. A PBR policy also can use an ACL to perform QoS mapping and marking for traffic that matches the 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 Layer 4 CAM on the interfaces and routes traffic that matches the ACLs according to the instructions in the route maps. You also can map and mark the traffic's QoS information using the QoS options of the ACLs.

Configuration Considerations

- JetCore supports an unlimited number of PBR policies that contain a single route map instance and a single ACL.
- JetCore supports up to 64 PBR policies that have more than one route map instance or more than one ACL. In this case, a given policy can have up to six route map instances, with up to six ACLs in each instance, and up to six next hops in each ACL.
- The ACL **log** and <icmp-type> options cause PBR to be performed by the CPU instead of in hardware. If you use either of these options in an ACL, no CAM entries are programmed for the ACL.
- 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 always selects the first next hop from the next hop list that is up, unless you use the **ip policy prefer-direct-route** option. If you use this option, PBR selects a direct route instead. 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 sends the traffic to the CPU for forwarding.
- For fragmented packets, by default PBR matches a fragment to an ACL if the source and destination addresses in the fragment exactly match an ACL. In this case, PBR uses the next hop that was used for the first fragment, which contains the Layer 4 UDP or TCP application port information. Alternatively, you can configure PBR to select the best next hop on an individual fragment basis.

NOTE: PBR is not supported for fragmented packets on 10 Gigabit Ethernet ports if the PBR's ACL filters on Layer 4 information. For 10 Gigabit Ethernet, the PBR policy sends fragmented packets on the Layer 3 paths.

Configuring a PBR Policy

To configure a PBR policy:

- Configure ACLs that contain the source IP addresses for the IP traffic you want to route using PBR. If you want to map or mark QoS information in the packets, use the QoS options in the ACLs.
- Configure a route map that matches on the ACLs and sets the route information.
- Optionally, enable PBR to use the most direct route if available.
- Apply the route map to an interface.

NOTE: If you are using software release 07.6.01 or later, enter the **ip rebind-acl** command at the global CONFIG level of the CLI to place ACL configuration changes into effect.

Configuration Examples

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.

```
BigIron(config)# access-list 101 permit tcp 10.10.10.0 0.0.0.255 eq http 5.5.5.0
0.0.0.255
BigIron(config)# route-map net10web permit 101
BigIron(config-routemap net10web)# match ip address 101
BigIron(config-routemap net10web)# set ip next-hop 1.1.1.1
BigIron(config-routemap net10web)# set ip next-hop 2.2.2.2
BigIron(config-routemap net10web)# exit
BigIron(config)# vlan 10
BigIron(config-vlan-10)# tagged ethernet 1/1 to 1/4

BigIron(config-vlan-10)# router-interface ve 1
BigIron(config)# interface ve 1
BigIron(config-vif-1)# ip policy route-map net10web
```

Next Hop Selection

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. An exception is if you use the **ip policy prefer-direct-route** option. In this case, the policy will instead use a direct route if available. If none of the policy's direct routes or next hops are available, PBR sends the traffic to the CPU for forwarding.

Using the Most Direct Route

To cause PBR policies to always use the most direct route available, enter the following command at the global CONFIG level of the CLI:

```
BigIron(config)# ip policy prefer-direct-route
```

Enabling PBR for Fragmented Packets

By default, PBR policies apply at Layer 3 only. The device matches traffic against the Layer 3 information in a PBR policy's ACLs, and applies the policy if the traffic matches the ACL. The device does not apply a PBR policy to a packet fragment even if the fragment's IP addresses match an ACL in the policy. Instead, the device forwards the fragment using a non-PBR route. This is true even if an ACL in a PBR policy contains Layer 4 information.

To apply a PBR policy to packet fragments:

- Add Layer 4 information to the PBR ACL.

- Enable fragment matching on the interface that has the PBR policy. You can enable fragment matching for the source Layer 4 port, destination Layer 4 port, or both.
  - Enable matching on the destination Layer 4 port in load balancing configurations where you want to ensure that traffic for a particular application is forwarded on the PBR path to the load balancers.
  - Enable matching on the source Layer 4 port if you want to ensure premium service for all traffic from a specific client.
  - Enable matching on both source and destination Layer 4 port if you want to ensure premium service for fragments in a given traffic flow.

The following example shows how to configure a PBR policy for Network File System (NFS) traffic, which uses UDP application port 2049. In this example, the next hop is selected individually for each fragment that exactly matches the destination IP address in one of the PBR policy's ACLs.

```
BigIron(config)# access-list 111 permit udp any host 2.3.3.5 eq 2049
BigIron(config)# route-map slbmap permit 1
BigIron(config-routemap slbmap)# match ip address 111
BigIron(config-routemap slbmap)# set next-hop 1.2.3.4
BigIron(config-routemap slbmap)# exit
BigIron(config)# interface ethernet 1/1
BigIron(config-if-1/1)# ip policy route-map slbmap
BigIron(config-if-1/1)# ip policy frag-match-dest
```

Creating a Route Map

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 up 50 route maps on the Layer 3 Switch.

The **permit | deny** parameter specifies the action the Layer 3 Switch 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. 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.

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.

NOTE: The **set ip default** option is not supported.

NOTE: The **set interface** option is not supported.

Creating ACLs

For detailed descriptions of the ACL syntax, see "Configuring Numbered and Named ACLs" on page 6-14.

Standard ACL

Syntax: [no] access-list <num> deny | permit <source-ip> | <hostname> <wildcard> [log]

or

Syntax: [no] access-list <num> deny | permit <source-ip>/<mask-bits> | <hostname> [log]

Syntax: [no] access-list <num> deny | permit host <source-ip> | <hostname> [log]

Syntax: [no] access-list <num> deny | permit any [log]

NOTE: If you use the **log** option, the ACL entry is sent to the CPU for processing.

Syntax: [no] ip access-group <num> in | out

NOTE: The **out** option is not supported in the hardware-based ACL mode.

Extended ACL

Syntax: [no] access-list <num> deny | permit <ip-protocol> <source-ip> | <hostname> <wildcard>
[<operator> <source-tcp/udp-port>] <destination-ip> | <hostname> [<icmp-type>] <wildcard>
[<operator> <destination-tcp/udp-port>]
[established]
[precedence <name> | <num>]
[tos <num>]
[priority 0 | 1 | 2 | 3]
[priority-mapping <8021p-value>]
[dscp-mapping <dscp-value>]
[dscp-marking <dscp-value>]
[log]

NOTE: The **priority**, **priority-mapping**, **dscp-mapping**, and **dscp-marking** options are supported in 07.6.01 and later and apply only to JetCore devices and to 10 Gigabit Ethernet modules. See “QoS Options for IP ACLs (Rule-Based ACLs)” on page 6-44.

Syntax: [no] access-list <num> deny | permit host <ip-protocol> any any [log]

NOTE: If you use the <icmp-type> or **log** option, the ACL entry is sent to the CPU for processing.

Syntax: [no] ip access-group <num> in | out

NOTE: The **out** option is not supported in the hardware-based ACL mode.

Creating a PBR Policy

Syntax: [no] ip policy route-map <map-name>

This command identifies a route map used by the PBR policy.

Syntax: [no] ip policy prefer-direct-route

This command configures the PBR policy to prefer a direct route when available.

Syntax: [no] ip policy frag-match-dest

This command configures the PBR policy to match on the destination Layer 4 port information as well as on the IP address information in the route map ACLs.

Syntax: [no] ip policy frag-match-src

This command configures the PBR policy to match on the source Layer 4 port information as well as on the IP address information in the route map ACLs.

Syntax: [no] ip policy frag-match-src-dest

This command configures the PBR policy to match on both the source and destination Layer 4 port information as well as on the IP address information in the route map ACLs.

Chapter 8
Configuring IronClad Rate Limiting (IronCore)

Foundry's IronClad rate limiting enables you to control the amount of bandwidth specific traffic uses on specific interfaces, by limiting the amount of data the interface receives or forwards for traffic. You can configure the following types of rate limiting:

- Fixed Rate Limiting – Enforces a strict bandwidth limit. The device forwards traffic that is within the limit but drops all traffic that exceeds the limit.
- 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.

NOTE: If you want to use ARP rate limiting, see "Rate Limiting ARP Packets" on page 12-44.

NOTE: To configure rate limiting on a JetCore Chassis device or the FastIron 4802, see "Configuring JetCore Rate Limiting (JetCore)" on page 9-1. To configure rate limiting on a FastIron Edge Switch (FES device), see "Configuring Rate Limiting on Other Foundry Devices" on page 11-1.

Rate limiting support differs depending on the Foundry product. Table 8.1 lists the devices with IronCore modules on which rate limiting is supported and the specific rate limiting support on each product.

Table 8.1: IronCore Rate Limiting Support in 07.6.01

Product	Type	Input				Output			
		Port	Port / VLAN	VLAN / VE	ACL	Port	Port / VLAN	VLAN / VE	ACL
BigIron or NetIron with VM1	Fixed	Y	N/A	N/A	N/A	Y	N/A	N/A	N/A
	Adapt[1]	Y	Y	Y[2]	Y[b3]	Y	Y	Y[b]	Y[bc]
BigIron with M2/M3/M4 (Layer 2 or Layer 3)	Fixed	Y	N/A	N/A	N/A	Y	N/A	N/A	N/A
	Adapt[4]	Y	N/A	N/A	Y[b]	Y[be]	N/A	N/A	Y[be]
NetIron with M4	Fixed	Y	N/A	N/A	N/A	Y	N/A	N/A	N/A
	Adapt	Y	N/A	N/A	Y[b5]	Y	N/A	N/A	Y[be]

Table 8.1: IronCore Rate Limiting Support in 07.6.01

Product	Type	Input				Output			
		Port	Port / VLAN	VLAN / VE	ACL	Port	Port / VLAN	VLAN / VE	ACL
FastIron II/II Plus/III (B2S image)	Fixed	Y	N/A	N/A	N/A	Y	N/A	N/A	N/A
	Adapt	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
FastIron II/II Plus/III (B2R, BL3 image)	Fixed	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
	Adapt	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A

1.The VM1 supports up to 48 input rate limiting policies and up to 48 output rate limiting policies.
2.Rate limiting for virtual routing interfaces does not apply to Layer 2 software.
3.MAC-based rate limiting also is supported.
4.A BigIron Chassis device with M2, M3, or M4 or a NetIron Chassis device with M4 supports up to 20 input rate limiting policies and up to 20 output rate limiting policies.
5.MAC-based rate limiting also is supported in Layer 3 code, but it is not supported in Layer 2 code.

Additional Notes

- Rate limiting is not supported on POS or ATM interfaces.
- If you configure Adaptive Rate Limiting and ACLs on the same port, rate limiting stops working on the port and only the ACLs take effect.
- Port-and-VLAN based rate limiting (Port / VLAN) is supported only on devices managed by the VM1.

Fixed Rate Limiting

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. The rate limit applies only to the direction you specify.

Fixed Rate Limiting applies to all types of traffic on the port.

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 send or receive the number of bytes you specify in the policy, but drops additional bytes.

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.

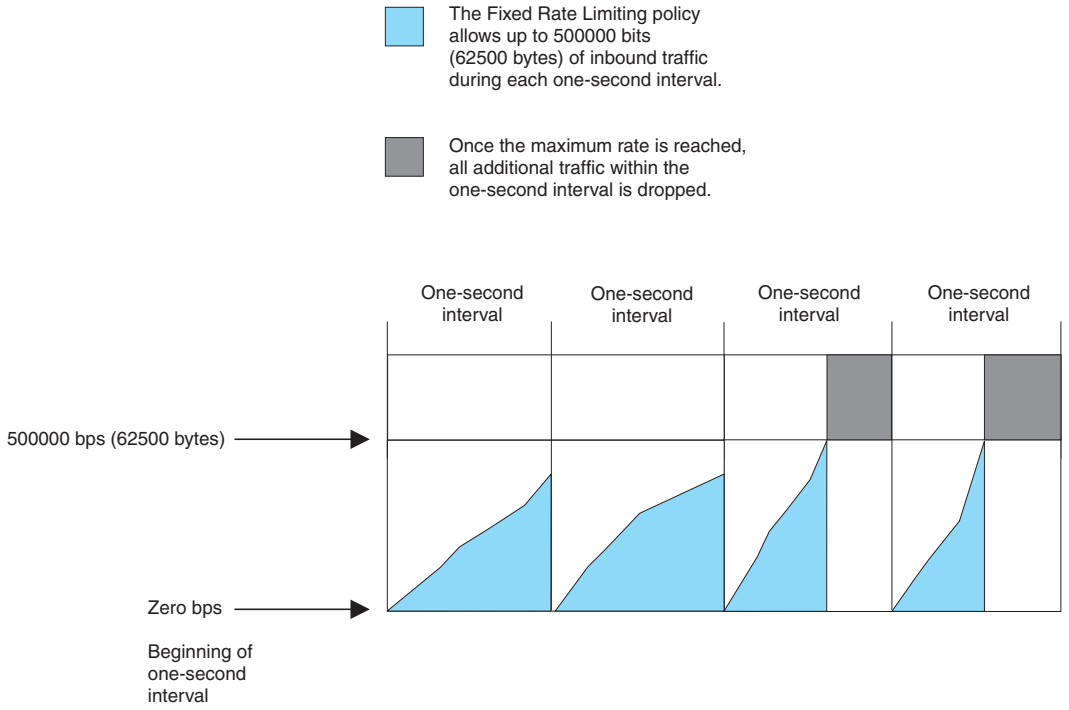
How Fixed Rate Limiting Works

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 8.1 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 8.1 Fixed Rate Limiting



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

To configure a Fixed Rate Limiting policy, enter a command such as the following at the configuration level for a port:

```
BigIron(config-if-1/1)# rate-limit input fixed 500000
```

This command configures a Fixed Rate Limiting policy that allows port 1/1 to receive a maximum of 500000 bps (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 | output fixed <rate>

The **input | output** parameter specifies whether the rate limit applies to inbound or outbound traffic on the port.

The <rate> parameter specifies the maximum rate for the port. Specify the rate in bits per second. You can specify from 1 up to any number. There is no default.

NOTE: If you specify a number that is larger than the port's line rate, the traffic will never cause the policy to go into effect.

Displaying Fixed Rate Limiting Information

To display configuration information and statistics for Fixed Rate Limiting, enter the following command at any level of the CLI:

```
BigIron(config)# show rate-limit fixed

Total rate-limited interface count: 6.
    Port     Input rate   RX Enforced    Output rate   TX Enforced
     1/1         500000             3
     2/1                                    1234567           100
     2/2                                    2222222             3
     2/3                                    1234567            15
     2/4                                    1238888            12
     2/5                                    1238888             7
```

Syntax: show rate-limit fixed

This display shows the following information.

Table 8.2: 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.
Input rate	The maximum rate allowed for inbound traffic. The rate is measured in bits per second (bps).
RX Enforced	The number of one-second intervals in which the Fixed Rate Limiting policy has dropped traffic received on the port.
Output rate	The maximum rate allowed for outbound traffic. The rate is measured in bps.
TX Enforced	The number of one-second intervals in which the Fixed Rate Limiting policy has dropped traffic queued to be sent on the port.

Adaptive Rate Limiting

The Adaptive Rate Limiting enables you to configure rate policies that enforce bandwidth limits for traffic. The features allows you to specify how much traffic of a given type a specific port can send or receive, and also allows you to either change the IP precedence of the traffic before forwarding it or drop the traffic.

You can apply rate policies to the following types of interfaces, in the inbound or outbound direction:

- Individual ports
- Trunk groups
- Virtual interfaces (used for routing by VLANs)
- Layer 2 port-based VLANs
- Port-and-VLAN based

You can apply up to 20 rate policy rules to an interface for inbound traffic and up to 20 more rules for outbound traffic. The interface can have up to 20 rules for each traffic direction. The device applies the rules in the order you apply them to the interface.

NOTE: JetCore Adaptive Rate Limiting applies only to IPv4 traffic.

NOTE: On Layer 2 devices and Layer 3 devices, you cannot apply rate limiting to a port if that port belongs to a VLAN that has a virtual interface. On Layer 3 devices, you cannot apply rate limiting to a port unless that port already has an IP address configured.

You can configure rate policies for the following types of traffic:

- Layer 3 IP traffic
- Specific source or destination IP addresses or networks
- Specific source or destination TCP or UDP application ports
- Specific MAC addresses

The rate policies you apply to an interface affect only the traffic types you specify and allows other traffic to be sent or received without rate limiting.

The rate policy rules allow to specify the action you want the Foundry device to take depending on whether the traffic is conforming to the policy. You can specify one of the following actions for each case:

- Forward the traffic
- Drop the traffic
- Change the IP precedence or the ToS value being used for a Diffserv control point, and forward the traffic
- Change the IP precedence or the ToS value being used for a Diffserv control point, then continue comparing the traffic to the rate policy rules
- Continue comparing the traffic to the rate policy rules without changing the IP precedence or Diffserv control point

NOTE: Foundry Adaptive Rate Limiting can change the value in the ToS field, which sometimes is used as a Diffserv code point. However, Foundry Adaptive Rate Limiting does not support RFC 2475.

The following sections provide examples of Adaptive Rate Limiting, an explanation of how the feature works, and configuration procedures.

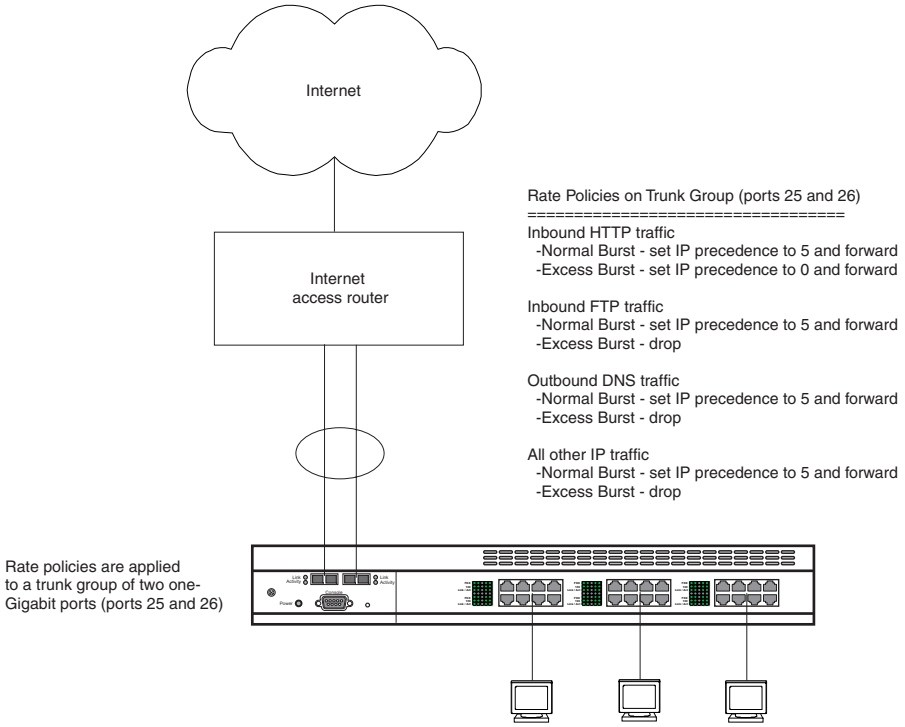
Examples of Adaptive Rate Limiting Applications

The following sections show some examples of how you can use Adaptive Rate Limiting. The CLI commands for implementing each application are shown in “Complete CLI Examples” on page 8-21.

Adaptive Rate Policies For a Trunk Group Uplink

Figure 8.2 shows an example of how you can use the Adaptive Rate Limiting. In this example, four rate policies are applied to the device’s uplink to the Internet. In this case, the uplink is a trunk group consisting of two one-Gigabit Ethernet ports.

Figure 8.2 Adaptive Rate Limiting applied to a Trunk Group Uplink



The rate policy rules are for three TCP/UDP applications: HTTP (web), FTP, and DNS. The fourth rule is for all other IP traffic (traffic that is not for one of the three applications). The device applies rate policy rules in the order in which you apply them to an interface. In this case, the rules are applied in the following order:

- Inbound HTTP traffic
- Inbound FTP traffic
- Outbound DNS traffic
- All other inbound IP traffic

Notice that each rule is associated with a traffic direction. You can apply a given rate policy rule to traffic received on an interface, sent on an interface, or both.

For each rule, the device counts the bytes that apply to the rule during each Committed Time Interval (time interval, which can be from 1/10th second up to one second). The device takes the conform action, which is action specified by the rule for Normal Burst Size, so long as the number of bytes for the traffic is within the Normal Burst Size value. Once the number of bytes exceeds the Normal Burst Size and thus enters the Excess Burst Size, the device takes the exceed action. “How Adaptive Rate Limiting Works” on page 8-10 describes how the byte counters for the Normal Burst Size and Excess Burst Size are incremented.

Each rule incudes one of the following actions depending on whether the traffic is conforming with the Normal Burst Size or has exceeded the Normal Burst Size:

- Forward the traffic
- Drop the traffic
- Change the IP precedence or the ToS value and forward the traffic
- Change the IP precedence or the ToS value, then continue comparing the traffic to the rate policy rules
- Continue comparing the traffic to the rate policy rules without changing the IP precedence or the ToS value

In Figure 8.2, all of the policies set the IP precedence to 5 (critical) for in traffic that conforms to the Normal Burst Size. In other words, for all packets up to the maximum number of bytes specified by the Normal Burst Size, the device sets the precedence in each packet to 5.

The policies take different actions for traffic in the Excess Burst Size. Some policies set the precedence and forward the traffic while other policies drop the traffic. In Figure 8.2, the rule for HTTP traffic sets the precedence to zero (routine) for traffic in the Excess Burst Size. The other policies drop the traffic.

In all cases, after the maximum number of bytes for the Normal Burst Interval and the Excess Burst Size match a given rule, the software drops additional bytes that match the rule until the burst size counters are reset.

Adaptive Rate Policy for a Virtual Routing Interface that Route VLANs

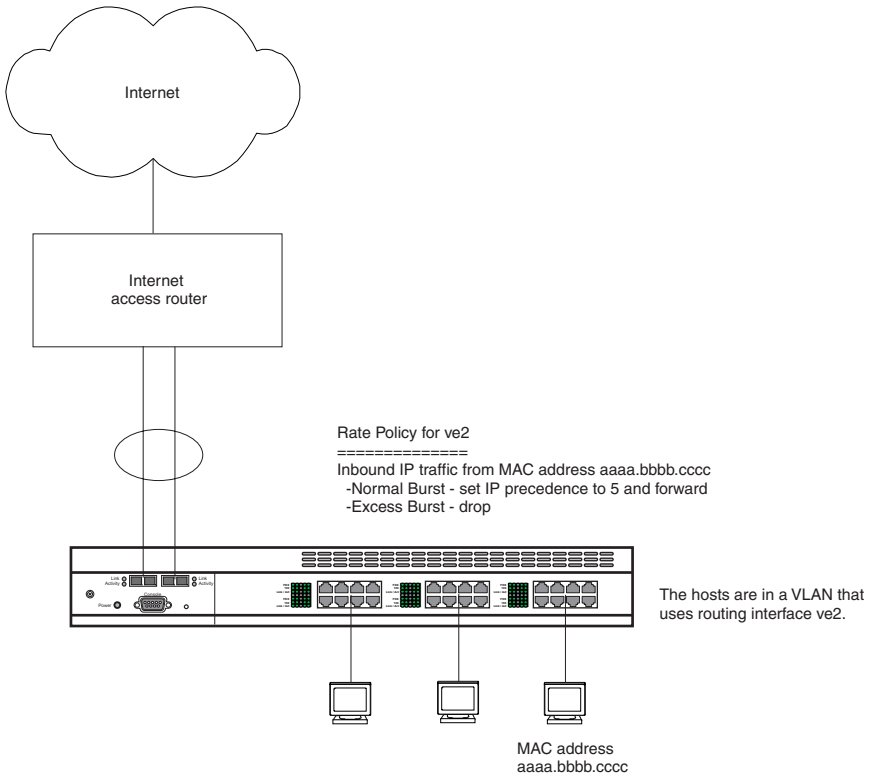
Figure 8.3 shows an example of a rate policy consisting of one rule applied to a virtual routing interface ("virtual interface" or "VE"). A virtual interface enables ports in a VLAN to route to other VLANs. In this example, the VLAN contains three ports, attached to three hosts. The hosts use virtual interface ve2 for routing.

Rate limiting policies for virtual routing interfaces that route VLANs can be applied only to the virtual routing interface and not on the physical port that is on the VLAN.

The rate limiting policy in this example forwards all conforming traffic from the host with MAC address aaaa.bbbb.cccc but drops all additional traffic from the host. Conforming traffic is traffic within the Normal Burst Size specified in the rate policy. Within a given Committed Time Interval, if the host sends more bytes than the number of bytes allowed by the Normal Burst Size, the policy drops the packets.

The other hosts in the VLAN do not have rules. As a result, their bandwidth is not limited.

Figure 8.3 Adaptive Rate Limiting applied to virtual routing interface that route VLANs



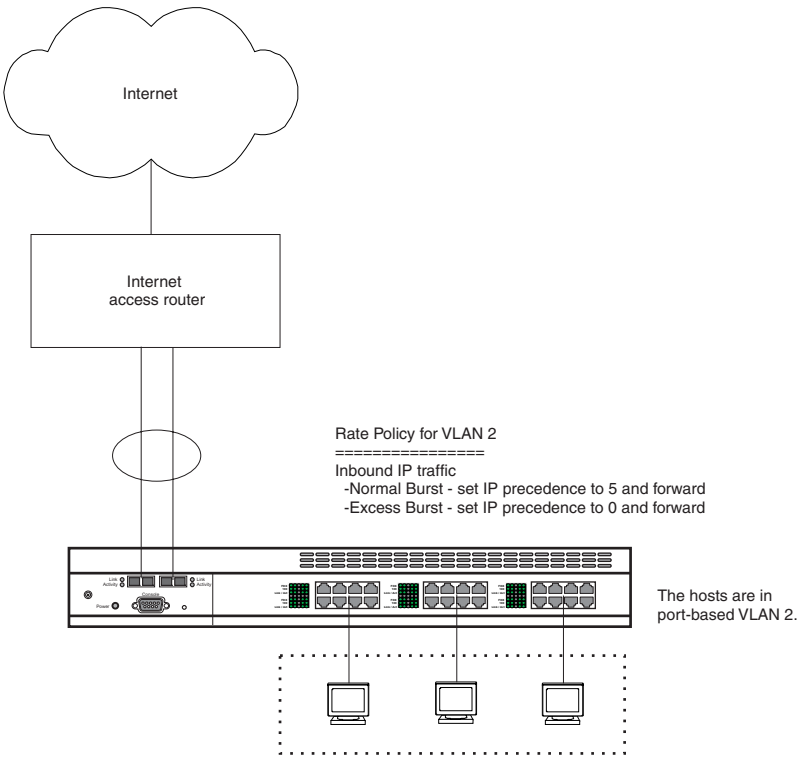
The rule could be applied to the port attached to the host for the same results. However, since the rule is associated with the virtual interface instead of a physical port, the policy remains in effect even if the host moves to another port within the VLAN.

Adaptive Rate Policy for a Layer 2 Port-Based VLAN

Figure 8.4 shows a rate policy applied to a VLAN. When you apply a rate policy to a VLAN, the policy applies to all the ports in the VLAN. The rate policy in this example performs the following actions on traffic received on ports in the VLAN:

- For conforming traffic, sets the precedence to 5
- For excess traffic, sets the precedence to 0

Figure 8.4 Adaptive Rate Limiting applied to a VLAN



NOTE: The rate policy in this example applies at Layer 2, while the policies in Figure 8.2 on page 8-6 and Figure 8.3 on page 8-7 apply at Layer 3. You cannot use ACLs for rate policies applied to directly to a VLAN. However, you can use ACLs if you apply the rate policy to a VLAN's virtual interface instead.

Adaptive Rate Limiting Parameters

The application examples in "Examples of Adaptive Rate Limiting Applications" on page 8-5 describe the rate policies but do not describe the parameters used to configure the policies. The parameters specify the portion of an interface's bandwidth you are allocating to specific traffic, the conforming and excess quantities of bytes for the traffic, and the granularity of the Adaptive Rate Limiting.

Adaptive Rate Limiting uses the following parameters:

- Average Rate
- Normal Burst Size
- Excess Burst Size
- Committed Time Interval

When you apply Adaptive Rate Limiting policies to an interface, you specify the first three of these parameters. The fourth parameter is derived from the first two.

NOTE: When you configure these parameters, express the Average Rate in bits. Express the Normal Burst Size and Excess Burst Size in bytes.

Average Rate

The Average Rate is a percentage of an interface's line rate (bandwidth), expressed as a number representing bits per second (bps). The value can be from 256Kbps up to the maximum line rate of the port. If the interface contains multiple ports (for example, a trunk group or a virtual interface), the maximum value is the combined line rate of all the ports in the interface.

Normal Burst Size

The Normal Burst Size is the maximum number of bytes that specific traffic can send on a port within the Committed Time Interval, and still be within that traffic's rate limit. The minimum value is 3277 or 1/10th of the Average Rate (whichever is higher), and the maximum value is the Average Rate.

Excess Burst Size

The Excess Burst Size is the upper threshold of the bandwidth you want to allow on the interface. In terms of bytes, it is the maximum number of additional bytes (bytes over the Normal Burst Size) within the Committed Time Interval that can be transmitted. The Excess Burst Size can be a value equal to or greater than the Normal Burst Size up to the maximum number of bytes the interface can forward within the Committed Time Interval (explained below).

NOTE: When you configure Adaptive Rate Limiting, to specify the Excess Burst Size you enter a value that is the sum of the Normal Burst Size and the number of bytes above the Normal Bust Size that you want to allow. For example, if you specify a Normal Burst size of 125000 and you want to allow up to 62500 additional bytes, specify the Excess Burst Size as 187500 (125000 + 62500).

Depending on how the rate limiting is configured, the device can take different actions for traffic within the Normal Burst Size and traffic that falls into the Excess Burst Size. For example, you can forward all traffic in the Normal Burst Size and reset the precedence to a lower priority for all Excess Burst Size traffic, or even just drop that traffic.

NOTE: Do not set the Excess Burst Size to a value greater than the maximum number of bytes the interface can forward within the Committed Time Interval. Even if the software allows you to specify a higher value, the interface cannot forward more data than its line rate supports.

Committed Time Interval

The Committed Time Interval is a value representing a slice of time on the interface where you apply the Adaptive Rate Limiting. The slice of time can be from 1/10th second up to one second. This parameter establishes the granularity of the Adaptive Rate Limiting. This parameter also determines the maximum value of the Excess Burst Size.

The Normal Burst Size counter increments during this slice of time, then reverts to zero when the next slice of time starts. The Excess Burst Time counter increments during every two Committed Time Intervals, then reverts to zero. See “How Adaptive Rate Limiting Works” on page 8-10.

The Committed Time Interval is not directly configurable, but is instead derived from the following formula:

- Normal Burst Size / Average Rate = Committed Time Interval

For example, you can configure parameters for a port as follows:

- Average Rate (in bits) = 10000000
- Normal Burst Size (in bytes) = 125000 (1000000 bits), which is 1/10th the Average Rate. 1/10th is the minimum value.

Thus, the Committed Time Interval is 1000000 bits / 10000000 bits = 0.1 seconds. This means that the Adaptive Rate Limiting parameters apply to time slices of bandwidth 0.1 seconds long.

To determine the maximum Excess Burst Size you can specify, use the Average Rate and Normal Burst Size you specified to calculate the Committed Time Interval. Then divide the interface's maximum line rate by the Committed Time Interval. Here are some examples:

- Assume that the interface is a 100Mbps port. The maximum line rate is therefore 100,000,000 bits per second, which is 12,500,000 bytes per second. Also assume that you specify an Average Rate of 40,000 bytes (320,000 bits / 8 = 40,000 bytes) and a Normal Burst Size of 4000 bytes. These values result in a Committed Time Interval of 0.1 (1/10th second). Multiply the interface's full line rate (12,500,000) by 0.1 to get 1,250,000. In this case, the maximum Excess Burst Size is 1250000 (1,250,000 bytes).
- Assume the same interface line rate, but specify an Average Rate of 80,000 bytes (640,000 bits / 8 = 80,000 bytes) and a Normal Burst Size of 8000 bytes. In this case, the Committed Time Interval is still 0.1 and the maximum Excess Burst Size is still 1,250,000 bytes.

Notice that in both of these examples, the Normal Burst Size is 1/10th the Average Rate, which in each case means the Committed Time Interval is 1/10th second. Because the interface's full line rate and the Committed Time Interval are the same in each case, the maximum Excess Burst Size is also the same in each case. However, the ratio of the Normal Burst Size to the Excess Burst Size in the examples is quite different.

NOTE: The Excess Burst Size, when entered as value during configuration, is the sum of the Normal Burst Size and the number of additional (excess) bytes you want to allow. For example, if you specify a Normal Burst size of 125000 and you want to allow up to 62500 additional bytes, specify the Excess Burst Size as 187500 (125000 + 62500).

How Adaptive Rate Limiting Works

Foundry's Adaptive Rate Limiting polices bandwidth usage on specific interfaces for specific IP traffic, and takes the actions you specify based on whether the traffic is within the amount of bandwidth you have allocated for the traffic or has exceeded the bandwidth allocation.

Adaptive Rate Limiting provides this service by counting the number of IP traffic bytes sent or received on an interface, then taking a specific action depending on whether the count is within the normal bandwidth allocation (Normal Burst Size) or has exceeded the allocation (Excess Burst Size).

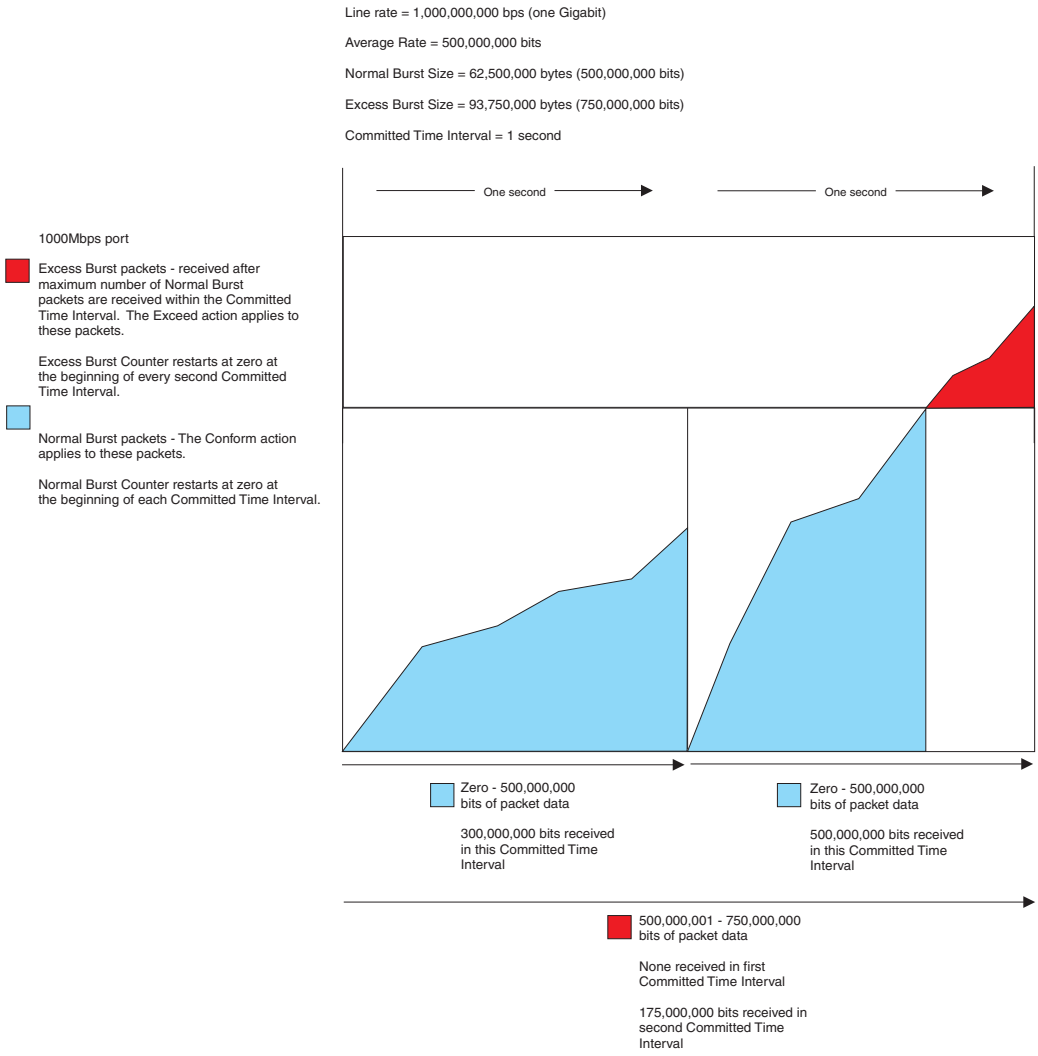
Normal Burst Size and Excess Burst Size Counters

The Adaptive Rate Limiting counts bytes within each Committed Time Interval, which is a slice of time (and thus a portion of the line rate) on the interface.

- Normal Burst Size counter – The byte counter for the Normal Burst Size increments during each Committed Time Interval, and is reset to zero at the next interval. Thus, the policy takes the action for conforming traffic for all the IP traffic's bytes up to the number of bytes specified by the Normal Burst Size.
- Excess Burst Size counter – The byte counter for the Excess Burst Size increments during each ***two*** Committed Time Intervals, and is reset to zero after every second interval. The policy takes the action for exceeding traffic for all the IP traffic's bytes past the maximum Normal Burst Size and up to the maximum Excess Burst Size. The device drops traffic once the number of bytes exceeds the maximum Excess Burst Size. The device continues dropping the packets until the next Committed Time Interval, at which time the Normal Burst Size is reset to zero.

Figure 8.5 shows an example of the Normal Burst Size and Excess Burst Size counters. This example shows two Committed Time Intervals.

Figure 8.5 Normal and Excess Burst Size Counters

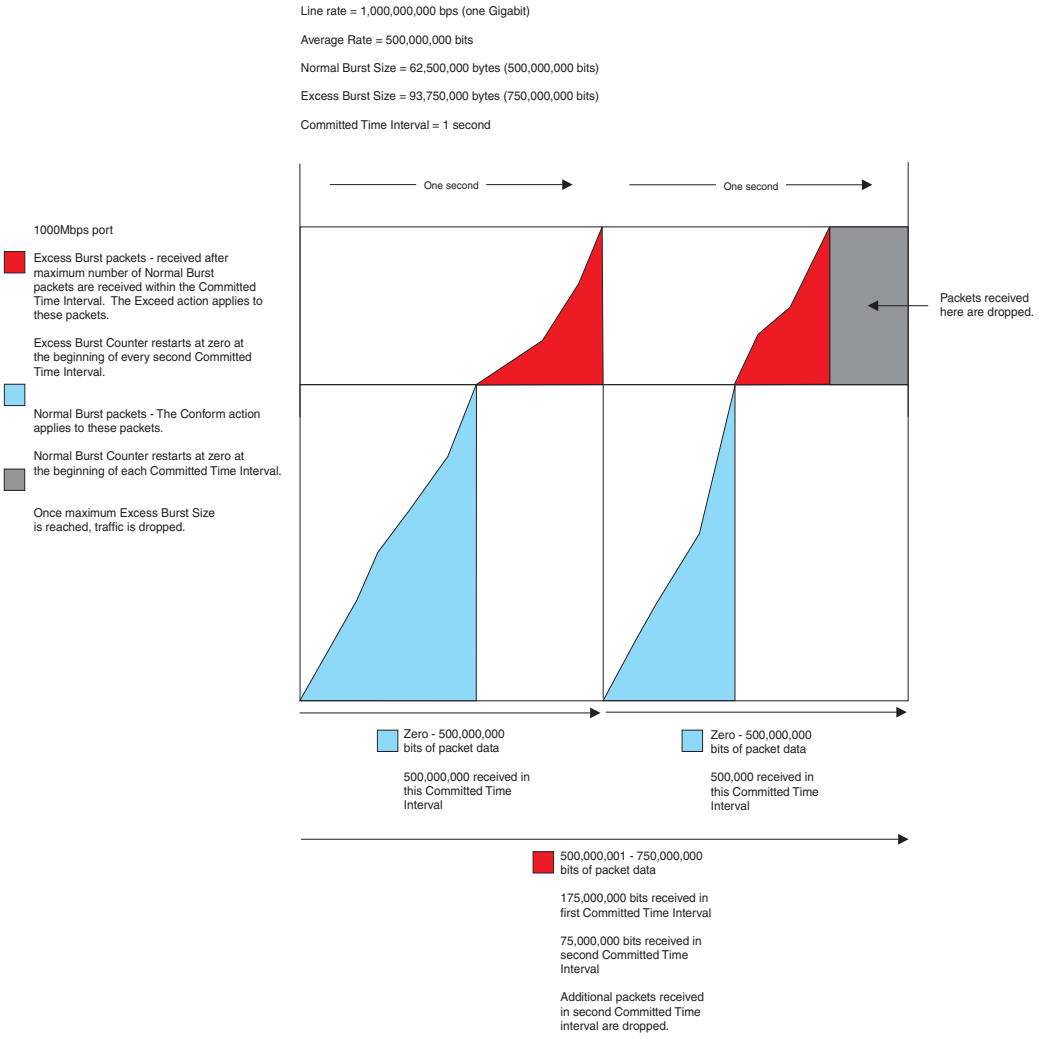


Notice that the counter for the Normal Burst Size counter restarts at the beginning of each Committed Time Interval, whereas the counter for the Excess Burst Size restarts after every two Committed Time Intervals. In this example, the policy rule on the interface matches 300,000,000 bits of IP traffic data during the first Committed Time Interval. Therefore, all the traffic conformed to the policy rule and the software took the action specified for conforming traffic.

During the second Committed Time Interval, the policy rule on the interface matches 675,000,000 bits of IP traffic data. Since the Normal Burst Size is 500,000,000, the software takes the conforming action for the first 500,000,000 bits. However, the software takes the exceed action for the remaining traffic. In this example, the action for conforming traffic is to set the IP precedence to 5, then forward the traffic. The action for exceed traffic is to set the IP precedence to 0, then forward the traffic.

Figure 8.6 shows an example of two Committed Time Intervals. In this example, the policy rule matches the maximum number of conforming bytes (Normal Burst Size bytes) in each interval.

Figure 8.6 Excess Burst Size increments over every two Committed Time Intervals



The rule matches additional bytes in each interval, and thus applies the exceed action. The counter for the Excess Burst Size increments over the span of the two intervals. Thus, the number of Excess Burst Size bytes available for the second interval is the amount that remains after the first Committed Time Interval. In this example, the rule matches 175,000,000 bits of additional (Excess Burst Size) data in the first Committed Time Interval. The Excess Burst Size in the rule is set to 250,000,000 bits. As a result, only 75,000,000 Excess Burst Size bits are available for use by the traffic that matches the rule in the second Committed Time Interval.

After the rule matches the maximum number of Normal Burst Size bytes in the second Committed Time Interval, the rule matches an additional 75,000,000 bits. The software drops all bytes received in the second Committed Time Interval after the Excess Burst Size maximum is reached.

Regardless of the actions for conforming and exceed traffic, the interface drops all traffic that matches a rule after the rule has matched the maximum number bytes for the rule's Normal Burst Size and Excess Burst Size.

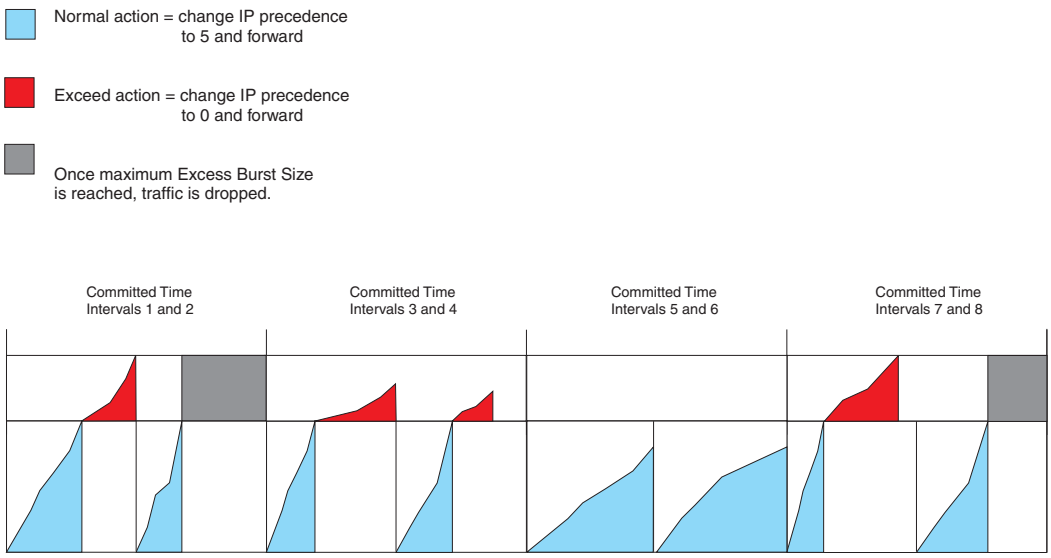
Figure 8.7 shows an example of eight Committed Time Intervals. The software drops traffic in the second and eighth intervals because the interface receives traffic that matches the rule after the rule has already matched the maximum number of bytes for the Normal Burst Size and Excess Burst Size.

In the third and fourth Committed Time Intervals, the rule matches the maximum number of bytes for the Normal Burst Size, and then matches additional bytes. However, the total number of excess bytes that match the rule over

these two Committed Time Intervals is not greater than the Excess Burst Size. Therefore, the software does not drop any of the matching traffic.

In the fifth and sixth Committed Time Intervals, the rule matches bytes but does not match even the maximum number of Normal Burst Size bytes in either interval. As a result, the rule does not need to apply the exceed action to any of the traffic that matches the rule in these intervals.

Figure 8.7 Traffic after the Excess Burst Size is reached is always dropped



Committed Time Interval

The Committed Time Interval specifies the granularity of the rate policing. The Committed Time Interval can be from 1/10th second up to one second. The length depends on the ratio of the Average Rate to the Normal Burst Size, parameters you specify when you configure a rate policy rule. The examples in the previous section all use a Committed Time Interval of one second. Since the Normal Burst Size is equal to the Average Rate, the ratio is 1:1. Therefore, the Committed Time Interval is one second.

The one-second interval is the least granular. The 1/10th-second interval is the most granular. To obtain the 1/10th-second interval, specify a Normal Burst Size that is 1/10th the Average Rate.

Configuring Adaptive Rate Limiting

To configure Adaptive Rate Limiting, perform the following steps:

- Characterize the traffic you want to manage. You can apply Adaptive Rate Limiting to any of the following:
  - All traffic (the default)
  - Traffic with certain precedence values sent or received on a specific interface
  - Traffic for specific source or destination IP host or network addresses
  - Traffic for specific TCP/UDP applications
  - Traffic from specific MAC addresses

  NOTE: To characterize the traffic, configure ACLs. You can use ACLs for rate policy rules applied to IP interfaces or to virtual interfaces, but not for rate policy rules applied directly to port-based VLANs. When you apply a rate policy rule to a port-based VLAN, the policy applies to all IP traffic.

- Specify how much bandwidth you want to allow the traffic for normal service, and whether you want the device to change the precedence for the traffic before forwarding it.

- For bandwidth above the normal service, specify the action you want the device to take. For example, you can configure the device to drop all traffic that exceeds the normal bandwidth allocation, or change the traffic's precedence or the ToS value, and so on.
- Apply the traffic characterization, the bandwidth limits, and the actions to incoming or outgoing traffic on a specific IP interface, virtual interface, or port-based VLAN.

NOTE: To configure port-, VLAN-, and direction-based rate limiting on a device managed by a VM1, see "Configuring Port-, VLAN- and Direction-Based Rate Limiting (VM1 only)" on page 8-18.

Characterizing the Traffic

You can use the following types of ACLs to characterize traffic. When you configure a rate policy rule on an interface, you can refer to the ACLs. In this case, the rate policy rule applies to the traffic that matches the ACLs.

- Standard IP ACL – Matches packets based on source IP address.
- Extended IP ACL – Matches packets based on source and destination IP address and also based on IP protocol information. If you specify the TCP or UDP IP protocol, you also match packets based on source or destination TCP or UDP application port.
- Rate limit ACL – Matches packets based on source MAC address, IP precedence or ToS values, or a set of IP precedence values.

You can configure a rate policy rule without using an ACL. In this case, the rule applies to all types of IP traffic. In fact, you cannot use ACLs in a rate policy rule you apply to a port-based VLAN. A rate policy rule you apply to a port-based VLAN applies to all types of IP traffic.

To configure the ACLs used by the rate policy in Figure 8.2 on page 8-6, enter the following commands:

```
BigIron(config)# access-list 101 permit tcp any any eq http
BigIron(config)# access-list 102 permit tcp any any eq ftp
BigIron(config)# access-list 103 permit udp any any eq dns
```

These ACLs match on all IP packets whose TCP application port is HTTP, FTP, or DNS.

To configure the rate limit ACL used in Figure 8.3 on page 8-7, enter the following command:

```
BigIron(config)# access-list rate-limit 100 aaaa.bbbb.cccc
```

The configuration in Figure 8.4 on page 8-8 applies a rate policy rule directly to a port-based VLAN and does not use ACLs.

Here is the syntax for standard ACLs.

Syntax: [no] access-list <num> deny | permit <source-ip> | <hostname> <wildcard> [log]

or

Syntax: [no] access-list <num> deny | permit <source-ip>/<mask-bits> | <hostname> [log]

Syntax: [no] access-list <num> deny | permit host <source-ip> | <hostname> [log]

Syntax: [no] access-list <num> deny | permit any [log]

NOTE: The **deny** option is not applicable to rate limiting. Always specify **permit** when configuring an ACL for use in a rate limiting rule.

Here is the syntax for extended ACLs.

Syntax: access-list <num> deny | permit <ip-protocol> <source-ip> | <hostname> <wildcard> [<operator> <source-tcp/udp-port>] <destination-ip> | <hostname> [<icmp-type>] <wildcard> [<operator> <destination-tcp/udp-port>] [precedence <name> | <num>] [tos <name> | <num>] [log]

NOTE: The **deny** option is not applicable to rate limiting. Always specify **permit** when configuring an ACL for use in a rate limiting rule.

Syntax: [no] access-list <num> deny | permit host <ip-protocol> any any [log]

NOTE: For complete syntax descriptions for standard and extended ACLs, see “Access Control List” on page 6-1.

Here is the syntax for rate limit ACLs.

Syntax: [no] access-list rate-limit <num> <mac-addr> | <precedence> | mask <precedence-mask>

The <num> parameter specifies the ACL number. Enter a value from 1–99 for a standard ACLs or a value from 100–199 for extended ACLs.

The <mac-addr> | <precedence> | **mask** <precedence-mask> parameter specifies a MAC address, an IP precedence, or a mask value representing a set of IP precedence values or a ToS value.

To specify a MAC address, enter the address in the following format: xxxx.xxxx.xxxx.

To specify an IP precedence, specify one of the following:

- **0** – The ACL matches packets that have the routine precedence.
- **1** – The ACL matches packets that have the priority precedence.
- **2** – The ACL matches packets that have the immediate precedence.
- **3** – The ACL matches packets that have the flash precedence.
- **4** – The ACL matches packets that have the flash override precedence.
- **5** – The ACL matches packets that have the critical precedence.
- **6** – The ACL matches packets that have the internetwork control precedence.
- **7** – The ACL matches packets that have the network control precedence.

To specify a mask value for a set of IP precedence values, enter **mask** followed by a two-digit hexadecimal number for the precedence values.

The precedence values are in an 8-bit field in the IP packet header. To calculate the hexadecimal number for a combination of precedence values, write down the values for the entire field to create the binary number for the mask value, then convert the number to hexadecimal. For example, to specify a mask for precedences 2, 4, and 5, write down the following values for the precedence field:

Bit position	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**
Precedence	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit pattern	0	0	1	1	0	1	0	0

Then, reading the digits from right to left, convert the number to hexadecimal. In this case, 00110100 binary becomes 0x34. Enter the mask as **mask 34**.

For simplicity, you can convert the digits in groups of four bits each.

For example, you can convert bits 1 – 4 (binary 0100) to get hexadecimal “4” for the right digit. Then convert bits 5 – 8 (binary 0011) to get hexadecimal “3” for the left digit. The result is “34”.

Alternatively, you can enter the entire eight-bit binary number in a calculator, then convert the number to hexadecimal. For example, you can enter the binary number “00110100” and convert it to hexadecimal to get “34”. (Without the leading zeros, enter “110100”.)

NOTE: The bits appear in this order in the IP precedence field and the software reads them from right to left. The least significant digit is the rightmost digit (bit position 1) and the most significant digit is the leftmost digit (bit position 8).

You also can use the **mask** <precedence-mask> parameter to specify a ToS value being used as a Diffserv control point. Regardless of whether the mask value you specify represents a set of IP precedences or a ToS value, the software examines the value in the field and responds with the action you specify.

Specifying the Bandwidth Allowances and Applying Rate Policy Rules to an Interface

When you apply a rate policy rule to an interface, you specify the following:

- The amount of the interface's bandwidth you are allowing for traffic that matches the rule
- The actions you want the device to take for traffic that conforms to the rule (is within the Normal Burst Size) and for traffic that exceeds the rule (is within the Excess Burst Size).

You can apply up 20 rate policy rules to an interface for inbound traffic and up to 20 additional rules for outbound traffic. The maximum number of rules for either direction is 20. When you apply more than one rule to an interface, the software interprets the rules in order, beginning with the first rule you apply to the interface and ending with the last rule you apply. When the traffic matches a rule, the software performs the action associated with that rule.

You can apply rate policy rules to the following types of interfaces:

- Physical port
- Trunk group (apply the policy to the trunk group's primary port)
- Virtual interface
- Port-based VLAN

CLI Examples

To specify the values for the rate policies in Figure 8.2 on page 8-6 and apply the policies, enter the following commands:

```
BigIron(config)# interface ethernet 1/25
BigIron(config-if-e1000-1/25)# rate-limit input access-group 101 10000000 125000
187500 conform-action set-prec-transmit 5 exceed-action set-prec-transmit 0
BigIron(config-if-e1000-1/25)# rate-limit input access-group 102 10000000 125000
187500 conform-action set-prec-transmit 5 exceed-action drop
BigIron(config-if-e1000-1/25)# rate-limit output access-group 103 1000000 100000
100000 conform-action set-prec-transmit 5 exceed-action drop
BigIron(config-if-e1000-1/25)# rate-limit input 4000000 80000 120000 conform-action
set-prec-transmit 5 exceed-action drop
```

To specify the values for the rate policies in Figure 8.3 on page 8-7 and apply the policies, enter the following commands:

```
BigIron(config)# interface virtual ve2
BigIron(config-ve-2)# rate-limit input access-group ratelimit 100 4000000 320000
400000 conform-action transmit exceed-action drop
```

To specify the values for the rate policies in Figure 8.4 on page 8-8 and apply the policies, enter the following commands:

```
BigIron(config)# vlan 2
BigIron(config-vlan-2)# rate-limit input 10000000 125000 187500 conform-action
set-prec-transmit 5 exceed-action set-prec-transmit 0
```

CLI Syntax

Syntax: [no] rate-limit input | output [access-group <num>] <average-rate> <normal-burst-size> <excess-burst-size> conform-action <action> exceed-action <action>

The **input | output** parameter specifies whether the rule applies to inbound traffic or outbound traffic.

- Specify **input** for inbound traffic.
- Specify **output** for outbound traffic.

The **access-group** <num> parameter specifies an ACL. When you use this parameter, the rule applies only to traffic that matches the specified ACL. Otherwise, the rule applies to all IP traffic that does not match a previous rule on the interface. You can specify the number of a standard ACL, and extended ACL, or a rate limit ACL. If you specify a rate limit ACL, use the parameter **ratelimit** (without a space) in front of the ACL number; for example, **ratelimit 100**.

NOTE: You cannot specify a named ACL.

The <average-rate> parameter specifies the portion, in bits per second (bps) of the interface's total bandwidth you want to allocate to traffic that matches the rule. You can specify a value can from 262144 (256Kbps) up to the maximum line rate of the port. For example, for a 100Mbps port, the maximum value is 100,000,000 (100Mbps).

If the interface is a trunk group, a virtual interface, or a VLAN, you can specify a value up to the maximum combined line rate of all the ports in the interface. For example, if the interface is a trunk group that consists of two one-Gigabit Ethernet ports, then the maximum value for <average-rate> is 2,000,000,000 (two times the maximum for each of the individual Gigabit ports).

The <normal-burst-size> parameter specifies the maximum number of bytes that specific traffic can send on the interface within the Committed Time Interval and still be within that traffic's rate limit. The minimum value is 3277[1] or 1/10th of the Average Rate (whichever is higher), and the maximum value is the Average Rate. The smallest fraction of the Average Rate you can specify is 1/10th.

The <excess-burst-size> parameter specifies the maximum number of additional bytes (bytes over the <normal-burst-size>) that can be transmitted within the Committed Time Interval. The <excess-burst-size> can be a value equal to or greater than the <normal-burst-size> up to the maximum number of bytes the interface can forward within the Committed Time Interval (see "Committed Time Interval" on page 8-9).

The device can take different actions for traffic within the <normal-burst-size> and traffic that falls into the <excess-burst-size>. For example, you can forward all traffic in the <normal-burst-size> and reset the precedence to a lower priority for all <excess-burst-size> traffic, or even just drop that traffic.

NOTE: Do not set the <excess-burst-size> parameter to a value greater than the maximum number of bytes the interface can forward within the Committed Time Interval. Even if the software allows you to specify a higher value, the interface cannot forward more data than its line rate supports.

The **conform-action** <action> parameter specifies the action you want the device to take for traffic that matches the rule and is within the Normal Burst Size. You can specify one of the following actions:

- **transmit** – Send the packet.
- **set-prec-transmit** <new-prec> – Set the IP precedence, then send the packet. You can specify one of the following:
  - **0** – routine precedence
  - **1** – priority precedence
  - **2** – immediate precedence
  - **3** – flash precedence
  - **4** – flash override precedence
  - **5** – critical precedence

1.This value comes from dividing the minimum Average Rate (262144 bits) by eight to get 32768 bytes, then dividing 32768 bytes by 10 to get 3276.8, since the smallest fraction of the Average Rate you can specify is 1/10th. The value 3276.8 is then rounded up to 3277.

    - **6** – internetwork control precedence
    - **7** – network control precedence
- **set-prec-continue** <new-prec> – Set the IP precedence to one of the values listed above, then evaluate the traffic based on the next rate policy.
- **drop** – Drop the packet.
- **continue** – Evaluate the traffic based on the next rate policy.

The **exceed-action** <action> parameter specifies the action you want the device to perform for traffic that matches the rule but exceeds the <normal-burst-size> within a given Committed Time Interval. You can specify one of the actions listed above.

Configuring Port-, VLAN- and Direction-Based Rate Limiting (VM1 only)

Adaptive port-and-VLAN based rate limiting is supported only on VM1. It allows you to rate-limit traffic on a given port, based on its VLAN tag. For example, if a port belongs to 3 VLANs, you can rate-limit each VLAN's traffic independently.

On Chassis devices managed by a VM1, you can configure a rate limiting policy for the following combination:

- Port
- VLAN ID
- Traffic direction

You can limit the rate on a specific port for a specific VLAN and for a specific traffic direction.

NOTE: You can use port-, VLAN-, and direction-based rate limiting for IP, AppleTalk, and IPX traffic. The other Adaptive Rate Limiting features described in this chapter apply only to IP traffic.

Configuration Considerations

- You can enable the feature on an individual port basis only. You cannot enable the feature on a virtual routing interface basis. This is true even if you have assigned a virtual routing interface to the trunk ports.
- The port on which you enable the feature cannot be a member of a virtual routing interface.
- When you enable the feature on a port, the following features are disabled on the port:
    - ACLs
    - Other Adaptive Rate Limiting features (for example, port-based or VLAN-based Adaptive Rate Limiting)
    - NetFlow
    - sFlow Export
    - Network Address Translation (NAT)
    - Policy-Based Routing (PBR)

    The configuration information for these features remains in the device's configuration but the features are disabled on the port.

Configuring a Port-, VLAN-, and Direction-Based Rate Limiting Policy

To configure a port-, VLAN-, and direction-based rate limiting policy, enter a command such as the following at the configuration level for the port:

```
BigIron(config-if-e100-3/8)# rate in vlan 10 262144 3277 3277 conform-action
transmit exceed-action drop
```

This command configures a rate limiting policy for inbound traffic to VLAN 10 on port 3/8. The policy transmits traffic that conforms with the specified rate (262144 bps) and drops traffic that exceeds the rate.

Syntax: [no] rate-limit input | output vlan <vlan-id> <average-rate> <normal-burst-size> <excess-burst-size> conform-action <action> exceed-action <action>

The **input | output** parameter specifies whether the rule applies to inbound traffic or outbound traffic.

- Specify **input** for inbound traffic.
- Specify **output** for outbound traffic.

The **vlan** <vlan-id> parameter specifies the VLAN ID. The rate limiting policy applies only to traffic to or from the specified VLAN ID. The policy does not apply to traffic to or from other VLANs on this port.

The <average-rate> parameter specifies the portion, in bits per second (bps) of the interface's total bandwidth you want to allocate to traffic that matches the rule. You can specify a value can from 262144 (256Kbps) up to the maximum line rate of the port. For example, for a 100Mbps port, the maximum value is 100,000,000 (100Mbps).

If the interface is a trunk group, a virtual interface, or a VLAN, you can specify a value up to the maximum combined line rate of all the ports in the interface. For example, if the interface is a trunk group that consists of two one-Gigabit Ethernet ports, then the maximum value for <average-rate> is 2,000,000,000 (two times the maximum for each of the individual Gigabit ports).

The <normal-burst-size> parameter specifies the maximum number of bytes that specific traffic can send on the interface within the Committed Time Interval and still be within that traffic's rate limit. The minimum value is 3277[1] or 1/10th of the Average Rate (whichever is higher), and the maximum value is the Average Rate. The smallest fraction of the Average Rate you can specify is 1/10th.

The <excess-burst-size> parameter specifies the maximum number of additional bytes (bytes over the <normal-burst-size>) that can be transmitted within the Committed Time Interval. The <excess-burst-size> can be a value equal to or greater than the <normal-burst-size> up to the maximum number of bytes the interface can forward within the Committed Time Interval. For information about the Committed Time Interval and the rate limiting algorithm, see "Adaptive Rate Limiting Parameters" on page 8-8.

The device can take different actions for traffic within the <normal-burst-size> and traffic that falls into the <excess-burst-size>. For example, you can forward all traffic in the <normal-burst-size> and reset the precedence to a lower priority for all <excess-burst-size> traffic, or even just drop that traffic.

NOTE: Do not set the <excess-burst-size> parameter to a value greater than the maximum number of bytes the interface can forward within the Committed Time Interval. Even if the software allows you to specify a higher value, the interface cannot forward more data than its line rate supports.

The **conform-action** <action> parameter specifies the action you want the device to take for traffic that matches the rule and is within the Normal Burst Size. You can specify one of the following actions:

- **transmit** – Send the packet.
- **drop** – Drop the packet.
- **continue** – Evaluate the traffic based on the next rate policy.

NOTE: The **set-prec-transmit** and **set-prec-continue** actions under **conform-action** <action> and **exceed-action** <action> are not supported.

The **exceed-action** <action> parameter specifies the action you want the device to perform for traffic that matches the rule but exceeds the <normal-burst-size> within a given Committed Time Interval. You can specify one of the actions listed above.

NOTE: The **access-group** <num> parameter is not supported. You cannot use an ACL with port-, VLAN-, and direction-based rate limiting.

1.This value comes from dividing the minimum Average Rate (262144 bits) by eight to get 32768 bytes, then dividing 32768 bytes by 10 to get 3276.8, since the smallest fraction of the Average Rate you can specify is 1/10th. The value 3276.8 is then rounded up to 3277.

Displaying Configuration Information and Statistics

To display the Adaptive Rate Limiting policies in effect on the device, and statistics for the policies, enter a command such as the following at any level of the CLI:

```
BigIron(config-if-e1000-1/1)# show interface ethernet 1/1 rate-limit
Input
matches: access-group 101
params: 10000000 bps, 125000 limit, 187500 extended limit
conform 0 packets, 0 bytes; action: set-prec-transmit 5
exceeded 0 packets, 0 bytes; action: set-prec-transmit 0
last packet: 0ms ago, current burst: 0 bytes
last cleared: 0 days 00:08:05 ago, conformed 0 bps, exceeded 0 bps
Output
matches: access-group 103
params: 1000000 bps, 100000 limit, 100000 extended limit
conform 0 packets, 0 bytes; action: set-prec-transmit 5
exceeded 0 packets, 0 bytes; action: drop
last packet: 0ms ago, current burst: 0 bytes
last cleared: 0 days 00:00:04 ago, conformed 0 bps, exceeded 0 bps
```

Syntax: show interface ethernet <portnum> | ve <num> rate-limit

Table 8.3: CLI Display of Adaptive Rate Limiting Information

This Field...	Displays...
matches	The Adaptive Rate Limiting policy
params	The policy parameters
last packet	The time that has elapsed since a packet matched the policy
current burst	The actual burst size at the time the software responded to the command to display this information
last cleared	The time when the statistics counters were last cleared using the **clear statistics** command
conformed	The number of packets that matched the policy and conformed with the normal burst size, since the statistics were last cleared
exceeded	The number of packets that matched the policy but exceeded the normal burst size, since the statistics were last cleared

Clearing Adaptive Rate Limiting Statistics

To clear Adaptive Rate Limiting statistics, enter a command such as the following:

```
BigIron# clear statistics rate-counters ethernet 1/1
```

This command clears the Adaptive Rate Limiting statistics that have been accumulated for port 1/1.

Syntax: clear statistics rate-counters [dos-attack | ethernet <portnum> | pos <portnum> | slot <slotnum>]

NOTE: The **dos-attack** parameter clears statistics about ICMP and TCP SYN packets dropped because burst thresholds were exceeded. See the “Protecting Against Denial of Service Attacks” chapter in the *Foundry Security Guide*.

Complete CLI Examples

This section lists and explains the CLI commands for implementing the Adaptive Rate Limiting applications in “Examples of Adaptive Rate Limiting Applications” on page 8-5.

Commands for Adaptive Rate Policies For a Trunk Group

To configure the Adaptive Rate Limiting application described in “Adaptive Rate Policies For a Trunk Group Uplink” on page 8-5, enter the following commands.

The first three commands configure extended ACLs to characterize the traffic. ACL 101 is for all web traffic. ACL 102 is for all FTP traffic. ACL 102 is for all DNS traffic. Each of the ACLs matches on any source and destination IP address.

```
BigIron(config)# access-list 101 permit tcp any any eq http
BigIron(config)# access-list 102 permit tcp any any eq ftp
BigIron(config)# access-list 103 permit udp any any eq dns
```

The following command changes the CLI to the configuration level for port 1/25. If the port is the primary port in a trunk group, the rate policy configuration applies to all ports in the trunk group. In this case, port 1/25 is the primary port in a trunk group that also contains port 1/26.

```
BigIron(config)# interface ethernet 1/25
```

The following command configures a rate limit rule that uses ACL 101.

```
BigIron(config-if-e1000-25)# rate-limit input access-group 101 10000000 125000
187500 conform-action set-prec-transmit 5 exceed-action set-prec-transmit 0
```

The rule compares all inbound packets on the trunk group to ACL 101. For packets that match the ACL, the rule either sets the IP precedence to 5 (critical) and then sends the packet, or sets the IP precedence to 0 (routine) and sends the packet. The rule sets the precedence to 5 for all packets received up to the maximum Normal Burst Size, 125000 bytes. Once the interface receives this many bytes in the inbound direction that match ACL 101, the device sets the precedence for the next 62500 bytes to the value associated with the Excess Burst Size.

The burst size counters increment for the duration of the Committed Time Interval, then change back to zero for the next Committed Time Interval. The length of the Committed Time Interval is determined by the ratio of the Average Rate to the Normal Burst Size. In this case, the ratio is 10:1, so the Committed Time Interval is 1/10th second long. The counter for the Normal Burst Size accumulates packets for 1/10th second, then returns to zero. The counter for the Excess Burst Size accumulates packets for 2/10ths second, then returns to zero.

The following command configures a rate limit rule that uses ACL 102. This rule also applies to inbound traffic. The action for packets that exceed the Normal Burst Size is different from the action in the rule above. The rule above sets the precedence to 0 in packets received after the maximum number of conforming packets (the number represented by the Normal Burst Size) is received within the Committed Time Interval.

The following rule drops packets received after the maximum number of conforming packets have been received.

```
BigIron(config-if-e1000-25)# rate-limit input access-group 102 10000000 125000
187500 conform-action set-prec-transmit 5 exceed-action drop
```

The following rule applies to traffic that matches ACL 103. Like the previous rule, this rule drops packets received after the maximum number of conforming packets have been received. However, notice that this rule applies to traffic in the outbound direction.

```
BigIron(config-if-e1000-25)# rate-limit output access-group 103 1000000 100000
100000 conform-action set-prec-transmit 5 exceed-action drop
```

The following command configures a rule for all IP traffic that does not match one of the ACLs used in the rules above.

```
BigIron(config-if-e1000-25)# rate-limit input 4000000 80000 120000 conform-action
set-prec-transmit 5 exceed-action drop
```


When you make configuration changes, make sure you save them to the startup-config file. If the system resets for any reason or you reload the software, the configuration changes you make are reinstated only if they have been saved to the startup-config file. Enter the following command to save configuration changes:

```
BigIron(config-if-e1000-25)# write memory
```

You can enter this command from any configuration level of the CLI.

Commands for Adaptive Rate Policy for a Virtual Routing Interface that Route VLANs

To configure the Adaptive Rate Limiting application described in "Adaptive Rate Policy for a Virtual Routing Interface that Route VLANs" on page 8-7, enter the following commands.

The following command configures a rate limit ACL to characterize the traffic. In this case, the rate policy is for a specific host, so the rate limit ACL specifies a host MAC address.

```
BigIron(config)# access-list rate-limit 100 aaaa.bbbb.cccc
```

The following command changes the CLI to the configuration level for virtual interface ve2.

```
BigIron(config)# interface virtual ve2
```

The following command configures rule for inbound traffic that matches the rate limit ACL configured above. The rule sends traffic that conforms to the Normal Burst Size and drops traffic received after the maximum number of conforming bytes have been received.

The Average Rate for the rule is 8000000 bps. The Normal Burst Size is 640000 bytes, and the Excess Burst Size is 800000 bytes. Based on the Average Rate and Normal Burst Size values, the Committed Time Interval is 6.4/10ths of a second, or about 2/3 seconds.

```
BigIron(config-ve-2)# rate-limit input access-group ratelimit 100 4000000 320000
400000 conform-action transmit exceed-action drop
```

The following command saves the configuration changes:

```
BigIron(config-ve-2)# write memory
```

Commands for Adaptive Rate Policy for a Layer 2 Port-Based VLAN

To configure the Adaptive Rate Limiting application described in "Adaptive Rate Policy for a Layer 2 Port-Based VLAN" on page 8-8, enter the following commands.

The following command changes the CLI to the configuration level for port-based VLAN 2.

```
BigIron(config)# vlan 2
```

The following command configures a rule for all inbound IP traffic on the VLAN's ports. The rule applies to all IP packets that come into the device on a port in VLAN 2.

```
BigIron(config-vlan-2)# rate-limit input 10000000 125000 187500 conform-action set-
prec-transmit 5 exceed-action set-prec-transmit 0
```

The following command saves the configuration changes:

```
BigIron(config-vlan-2)# write memory
```

Disabling Rate Limiting Exemption for Control Packets

By default, the Foundry device does not apply Adaptive Rate Limiting policies to certain types of control packets, but instead always forwards these packets, regardless of the rate limiting policies in effect.

NOTE: This section applies only to Adaptive Rate Limiting. Fixed Rate Limiting drops all packets that exceed the limit, regardless of packet type.

Table 8.4 lists the types of control packets that are exempt from rate limiting by default.

Table 8.4: IP Control Traffic Exempt from Rate Limiting

Traffic Type		IP Address	IP Protocol or Application Port
IP multicast	IP nodes multicast	224.0.0.1	
	IP routers multicast	224.0.0.2	
	IP DVMRP router multicast	224.0.0.4	
	IP OSPF router multicast	224.0.0.5	
	IP OSPF designated router multicast	224.0.0.6	
	IP RIP V.2 router multicast	224.0.0.9	
	IP VRRP multicast	224.0.0.18	
IP unicast	BGP control packet		TCP port 179 (0x00B3)
	OSPF control packet		IP protocol type 89 (0x59)
	RIP packet		UDP port 520 (0x0208)

To provide exemption, the CPU examines each packet to determine whether the packet is one of the exempt control types. If your network does not use these control types and you want to reduce CPU utilization, you can disable exemption for the control packets on an interface. To do so, use the following CLI method.

NOTE: If your network uses BGP, OSPF, or RIP and you disable exemption, the rate limiting polices can result in routing protocol traffic being dropped.

To disable rate limiting exemption for control packets on an interface, enter the following command at the CLI configuration level for that interface:

```
BigIron(config-if-e1000-25)# rate-limit control-packet no
```

This command disables exemption of all the control packets listed in Table 8.4 on port 25.

Syntax: [no] rate-limit control-packet no | yes

To re-enable exemption for the interface, enter the following command:

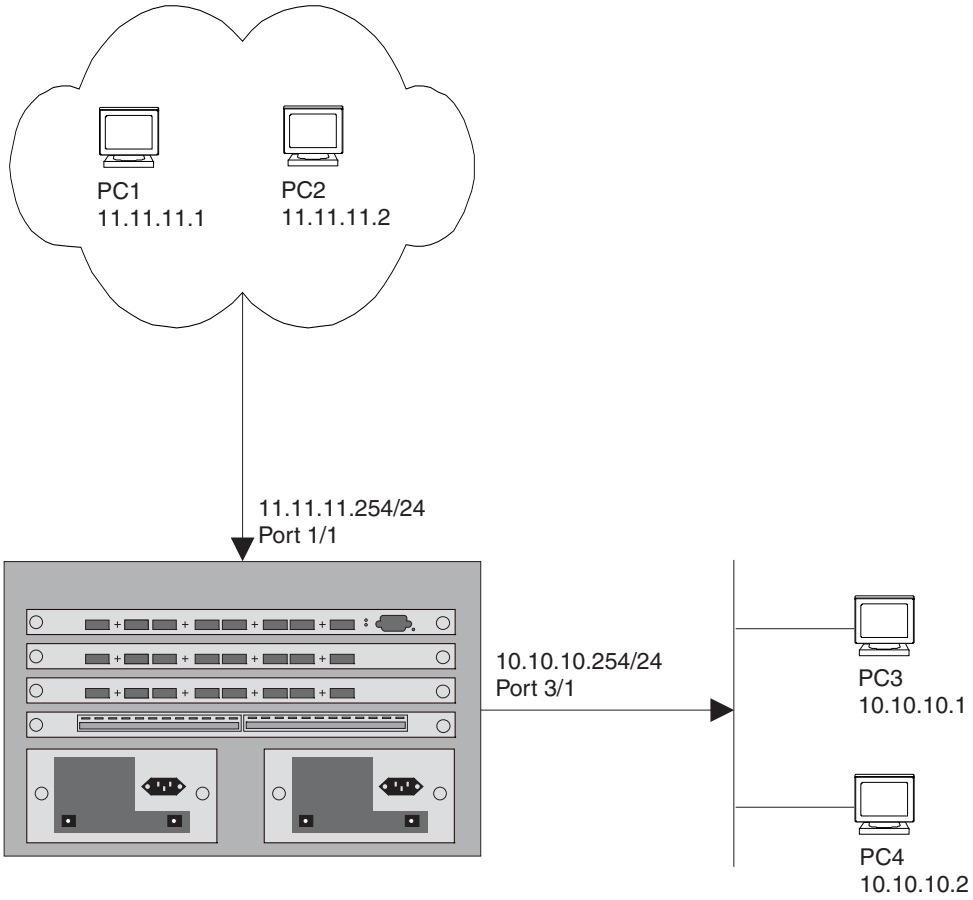
```
BigIron(config-if-e1000-25)# rate-limit control-packet yes
```

Using a Rate Limiting ACL to Deny Traffic

You cannot use filtering ACLs and rate limiting ACLs on the same port. However, you can use an ACL-based rate limiting policy to filter out (deny) IP traffic on a port in addition to other ACLs that limit the rate of the port.

Figure 8.8 shows an example of a configuration that uses rate limiting polices to limit the IP traffic from one host while denying IP traffic from the other host. Both hosts are attached to the Foundry device on the same port.

Figure 8.8 Filtering and rate limiting traffic on the same port



This configuration uses two rate limiting policies. The first policy limits the rate of IP traffic from PC1. The second policy drops IP traffic from PC2, by setting the conform and exceed actions both to drop.

Here are the CLI commands for this configuration.

```
BigIron(config)# access-list 1 permit host 11.11.11.1 log
BigIron(config)# access-list 2 permit host 11.11.11.2 log
BigIron(config)# interface ethernet 1/1
BigIron(config-if-e100-1/1)# rate-limit input access-group 1 262144 3277 3277
conform-action continue exceed-action drop
BigIron(config-if-e100-1/1)# rate-limit input access-group 2 262144 3277 3277
conform-action drop exceed-action drop
```

The first rate limiting policy limits the rate of traffic from PC1 (11.11.11.1). The policy forwards traffic that conforms to the policy's rate but drops traffic that exceeds the rate. The second rate limiting policy drops all IP traffic from PC2 (11.11.11.2). The policy uses the deny action for traffic that conforms to the rate or exceeds the rate.

NOTE: For this configuration to work correctly, the rate-limiting policy that denies all traffic must be the last policy you apply to the port.

Chapter 9
Configuring JetCore Rate Limiting (JetCore)

This chapter describes how to configure rate limiting on devices with JetCore modules. Rate Limiting on devices with JetCore modules can be one of the following types:

- Adaptive Rate Limiting for Chassis devices with JetCore modules and the FastIron 4802
- Fixed Rate Limiting for devices with JetCore modules running Service Provider software release 09.1.00

NOTE: To configure rate limiting on an IronCore module, see "Configuring IronClad Rate Limiting (IronCore)" on page 8-1. To configure rate limiting on a FastIron Edge Switch (FES device), see "Configuring Rate Limiting on Other Foundry Devices" on page 11-1.

Adaptive Rate Limiting

Line-rate rate limiting in hardware is available on the following devices:

- Foundry devices with JetCore modules running software release 07.6.01 and later
- FastIron 4802 devices running software release 07.6.01 and later

You can configure the device to use one of the following modes of rate limiting:

- Port-based – Limits the rate on an individual port to the maximum bits per second (bps) you specify.
- Port-and-priority-based – Limits the rate on an individual hardware forwarding queue on an individual port.
- ACL-based – Limits the rate for IP traffic on an individual port that matches the permit conditions in IP Access Control Lists (ACLs). You can use standard or extended IP ACLs. Standard IP ACLs match traffic based on source IP address information. Extended ACLs match traffic based on source and destination IP address and IP protocol information. For TCP and UDP, they also match on source and destination TCP or UDP addresses.
- JetCore Layer 2 ACL-based – This feature is an extension to the existing IP ACL-based rate limiting on devices with JetCore modules. This feature enables you to limit traffic rates using the Layer 2 parameters defined in the associated JetCore Layer 2 ACL table.

NOTE: Port-and-priority-based rate limiting, ACL-based rate limiting, and JetCore Layer 2 ACL-based rate limiting are supported only for inbound rate limiting policies. Port-based rate limiting is supported for inbound and outbound rate limiting policies.

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.

NOTE: The adaptive rate limiting described in this section is supported on JetCore modules and on the FastIron 4802. This rate limiting is not supported on devices with IronCore modules or on the 10 Gigabit Ethernet module.

JetCore Rate Limiting Support for Release 07.6.01

Table 9.1 lists the types of rate limiting supported on devices with JetCore modules, running software release 07.6.01.

Table 9.1: JetCore Rate Limiting Support in Devices Running Software Release 07.6.01

Product	Input			Output		
	Port	Port-and-priority	ACL	Port	Port-and-priority	ACL
BigIron 4000/8000/15000	Y	Y	Y[1]	Y	N/A	N/A
FastIron 400/800/1500	Y	Y	Y[a]	Y	N/A	N/A
FastIron 4802 (FWS4802 and FWS4802-PREM) (Layer 2 or Layer 3)	Y	Y	Y[a]	Y	N/A	N/A

1.ACL-based Adaptive Rate Limiting is supported on individual ports only, not on virtual routing interfaces. Up to ten ACL-based rate limiting policies are supported per port and up to 105 are supported per device. You cannot use the ACL-based mode along with features that modify the ToS value in IP traffic.

Additional Notes

- Rate limiting is not supported on POS or ATM interfaces.
- If you configure Adaptive Rate Limiting and ACLs on the same port, rate limiting stops working on the port and only the ACLs take effect.
- VLAN-based (VLAN / VE) rate limiting is not supported.
- Port-and-VLAN based rate limiting (Port / VLAN) is not supported.
- Rate limiting is not supported on FastIron devices with JetCore modules J-FIxGMR4-BASE or J-FIxG-BASE.
- FastIron 4802 only – You cannot use outbound rate limiting on 10/100 ports and a Gigabit Ethernet port at the same time, if the ports are managed by the same IPC. 10/100 ports 1 – 24 and Gigabit Ethernet port 49 are managed by IPC 1. 10/100 ports 25 – 48 and Gigabit Ethernet port 50 are managed by IPC 2. This issue does not apply to Chassis devices with JetCore modules.

Rate Limiting Algorithm and Parameters

Rate limiting uses the following algorithm:

c = (R * I * 0.0192) / (S * 8)

where:

- C is the number of ***Credits***. A policy allows up to the number of bytes for which the policy has credits in a given Rate Limiting Interval. The algorithm rounds the value of C up to the next whole integer. Inbound rate limiting uses 32-byte credits. Outbound rate limiting uses 64-byte credits.
- R is the ***Average Rate***. The Average Rate is the maximum number of bits the policy allows during one second. This parameter is configurable.
- I * 0.0000192 calculates the ***Rate Limiting Interval***. The Rate Limiting Interval determines the granularity of the rate limiting. The value of I depends on the type of rate limiting (inbound or outbound) and the port type. See Table 9.2.
- S is the credit size. Multiplying S by 8 converts bits to bytes, since the Average Rate is expressed in bits per second but the Credits are based on bytes.

The device calculates the Credits based on the Average Rate and Rate Limiting Interval.

Table 9.2 lists the rate limiting parameters.

Table 9.2: JetCore Rate Limiting Parameters

Traffic Direction	Port Type	Minimum Average Rate (R)	Rate Increments (Granularity)[1]	Time Interval (I)	Credit Size (S)
Inbound	10/100	256512 bps	256512 bps	52	32
	Gigabit	1025792 bps	1025792 bps	13	32
Outbound	10/100	1041910 bps	41500 bps	640	64
	Gigabit	20833792 bps	833024 bps	32	64

1.The rate increments are approximate.

The following sections describe the rate limiting parameters in detail.

Average Rate

The Average Rate is a parameter you specify when you configure a rate limiting policy. The Average Rate represents a percentage of an interface's line rate (bandwidth), expressed in bits per second (bps). The Average Rate specifies the maximum number of bits you want to allow a port to receive or forward during a one-second interval.

The Average Rate you can specify depends on the port's maximum line rate and whether you are configuring inbound rate limiting or outbound rate limiting. Table 9.2 lists the minimum Average Rate for each traffic direction and port type. The maximum Average Rate you can specify is the maximum line rate of the port.

Adjusted Average Rate

The software adjusts the Average Rate you enter so that the calculation of credits does not result in a remainder of a partial Credit. The CLI displays the adjusted rate. You also can display a table of the adjusted rate values. See "Displaying Adjusted Average Rates" on page 9-13.

For outbound rate limiting, it can take 30 – 60 seconds for a port's rate to change to the adjusted Average Rate. This can occur in the following cases:

- When you apply an outbound rate limiting policy to the port.
- When the packet sizes of the traffic change dramatically within a short period of time.

Credits

A Credit is a forwarding allowance for a rate-limited port, and is the smallest number of bytes that can be allowed during a given Rate Limiting Interval. Inbound rate limiting uses 32-byte credits. Outbound rate limiting uses 64-byte credits.

During a Rate Limiting Interval, a port can send or receive only as many bytes as the port has Credits for. For example, if an inbound rate limiting policy results in a port receiving two Credits per rate limiting interval, the port can send or receive a maximum of 64 bytes of data during that interval.

Rate Limiting Interval

The Rate Limiting Interval is a specific number of milliseconds (ms) that determines the granularity of the rate limiting. Table 9.2 lists the rate limiting interval Average Rate for each traffic direction and port type. JetCore Adaptive Rate Limiting allocates Credits on an individual Rate Limiting Interval basis.

Rate Limiting of Control Packets

For the port-based and port-and-priority-based modes, rate limiting applies to all packets including the following control packets. For the ACL-based mode, rate limiting does not apply to any of these control packets. Table 9.3 lists the types of control packets that are not rate limited for the ACL-based mode.

Table 9.3: IP Control Traffic Exempt from Rate Limiting when Using the ACL-based Mode

	MAC Address	IP Address	IP Protocol or Application Port
Layer 2 broadcast	FFFF.FFFF.FFFF		
Layer 2 multicast	0100.5E000.0000 – 0100.5E00.FFFF		
Layer 2 subnet directed broadcast	Any		
Layer 3 local multicast		E0.00.00.00 with mask E0.00.00.FF	
Layer 3 IGMP multicast		E0.00.01.00 – EF.FF.FF.FF	
PIM control packet			IP protocol 103
OSPF control packet			IP protocol 89
RIP packet			UDP port 520 (0x0208)
BGP control packet			TCP port 179 (0x00B3)

Configuration Considerations

- Inbound rate limiting and outbound rate limiting are completely independent of one another. You can configure rate limiting for either direction or both directions on the same port. However, for each traffic direction, there are some restrictions to the types of rate limiting you can use in combination for that traffic direction.

- For outbound rate limiting, you can use port-based rate limiting only. Port-and-priority based rate limiting and ACL-based rate limiting are not supported.
- For inbound rate limiting, Table 9.4 lists the types of rate limiting you can use together.

Table 9.4: Valid Inbound Rate Limiting Combinations

Rate Limiting Type	Can be Used Together?		
	Port	**Port-and-Priority**	**ACL**
Port		No	Yes
Port-and-Priority	No		No
ACL	Yes	No	

NOTE: There is one exception to the support for both port and ACL-based rate limiting for inbound traffic. You cannot use port-based rate limiting on the first port on an IGC or IPC if you have already applied an ACL-based rate limiting policy to another port on the same IGC or IPC. If you want to use both types of rate limiting on ports managed by the same IGC or IPC, including the first port managed by the IGC or IPC, use an ACL-based rate limiting policy on the first port. You can then use port-based or ACL-based rate limiting policies on any of the other ports managed by the IGC or IPC.

- JetCore hardware-based rate limiting is not supported on trunk groups.
- For inbound traffic, you can use port-based or port-and-priority-based rate limiting on a port that is a member of a VLAN that has a virtual routing interface.
- If you use the ACL-based mode, by default the device forwards traffic that matches the deny conditions in the ACLs you use in the rate limiting policies on the port. However, you can configure the device to drop this traffic instead. See “Using ACLs for Filtering in Addition to Rate Limiting” on page 9-9.
- You cannot use the ACL-based mode along with features that modify the Type-of-Service (ToS) value in IP traffic. For traffic that matches the permit conditions in a rate limiting ACL, the device leaves the ToS values unchanged even if other features on the device are configured to change the ToS values.
- If you configure an ACL-based rate limiting policy, the device sets the TCP and UDP ACL modes to strict TCP and non-strict UDP. These modes are required to create the CAM entries for rate limited traffic. When you are not using ACLs for rate limiting, the modes affect processing for ACL-based filtering.

  NOTE: You cannot change the setting from strict TCP or non-strict UDP unless you remove the rate limiting policies first.

- Software-based rate limiting (the type supported on Chassis devices with IronCore modules) is not supported.

NOTE: See also “JetCore Rate Limiting Support for Release 07.6.01” on page 9-2.

Configuring JetCore Adaptive Rate Limiting

The following sections show examples for configuring rate limiting policies for each mode and describe the CLI syntax.

In each example, the CLI adjusts the Average Rate you enter to be valid for the Credit calculation. To display a table of adjusted Average Rates, see “Displaying Adjusted Average Rates” on page 9-13.

Configuring a Port-Based Rate Limiting Policy

To configure an inbound port-based policy, enter commands such as the following:

```
BigIron(config)# interface ethernet 1/1
BigIron(config-if-e100-1/1)# rate-limit in 600000
The average rate has been adjusted to 513024
```

These commands configure an inbound policy on 10/100 Ethernet port 1/1 with an Average Rate of 513024 bps. The following commands configure an inbound rate limiting policy on a Gigabit Ethernet port.

```
BigIron(config)# interface ethernet 2/1
BigIron(config-if-e1000-2/1)# rate-limit in 2000000
The average rate has been adjusted to 2051328
```

To configure an outbound port-based policy, enter commands such as the following:

```
BigIron(config)# interface ethernet 1/2
BigIron(config-if-e100-1/2)# rate-limit out 5000000
The average rate has been adjusted to 5000192
```

These commands configure an outbound policy on 10/100 Ethernet port 1/2 with an Average Rate of 5000192 bps. The following commands configure an outbound rate limiting policy on a Gigabit Ethernet port.

```
BigIron(config)# interface ethernet 2/2
BigIron(config-if-e1000-2/2)# rate-limit out 40000000
The average rate has been adjusted to 40000512
```

Syntax: [no] rate-limit in | out <average-rate>

See “Rate Limiting Syntax” on page 9-7.

Configuring a Port-and-Priority-Based Rate Limiting Policy

Port-and-priority-based rate limiting is supported for inbound traffic only. To configure a port-and-priority-based policy, enter commands such as the following:

```
BigIron(config)# interface ethernet 1/1
BigIron(config-if-e100-1/1)# rate-limit in priority q0 q2 600000
The average rate has been adjusted to 513024
```

These commands configure an inbound policy on 10/100 Ethernet port 1/1, for hardware forwarding queues q0 and q2 with an Average Rate of 769280 bps. The policy applies only to traffic that is received on the port and is placed in the specified forwarding queues.

Syntax: [no] rate-limit in priority q0 | q1 | q2 | q3 <average-rate>

See “Rate Limiting Syntax” on page 9-7.

Configuring an ACL-Based Rate Limiting Policy

You can use standard or extended IP ACLs for ACL-based rate limiting. ACL-based rate limiting is supported for inbound traffic only.

- Standard IP ACLs match traffic based on source IP address information.
- Extended ACLs match traffic based on source and destination IP address and IP protocol information. For TCP and UDP, they also match on source and destination TCP or UDP addresses.

NOTE: If you apply an ACL-based rate limiting policy to a port that belongs to a virtual routing interface, by default the policy applies only to routed traffic, not to traffic switched among ports within the VLAN.

To configure ACL-based policies on a port, enter commands such as the following:

```
BigIron(config)# access-list 50 permit host 1.1.1.2
BigIron(config)# access-list 60 permit host 2.2.2.3
BigIron(config)# interface ethernet 1/1
BigIron(config-if-e100-1/1)# rate-limit in access-group 50 600000
The average rate has been adjusted to 513024
BigIron(config-if-e100-1/1)# rate-limit in access-group 60 3000000
The average rate has been adjusted to 3077120
```

These commands configure two inbound rate limiting policies on 10/100 Ethernet port 1/1. The first policy rate limits traffic from IP host 1.1.1.2. The second policy rate limits traffic from IP host 2.2.2.3.

NOTE: Use the **permit** condition for traffic that you want to include in the policy. If you use the **deny** condition, the policy does not apply to the specified traffic. Depending on whether the strict ACL option is enabled, the device either forwards denied traffic without rate limiting it, or drops the traffic. See “Using ACLs for Filtering in Addition to Rate Limiting” on page 9-9.

NOTE: You must configure the ACLs before you can use them to configure the rate limiting policy.

Syntax: [no] rate-limit in access-group <acl-id> <average-rate>

See “Rate Limiting Syntax”.

Rate Limiting Syntax

Syntax: [no] rate-limit in | out
[[priority q0 | q1 | q2 | q3] | [access-group <acl-id>]]
<average-rate>

The **in | out** parameter specifies the traffic direction to which the policy applies.

NOTE: The **out** option is supported only for port-based rate limiting. The **out** option is not supported for port-and-priority-based rate limiting or ACL-based rate limiting.

The syntax allows you to configure a port-based policy, a port-and-priority-based policy, or an ACL-based policy.

- To create a port-based policy, do not use the **priority** or **access-group** parameters.
- To create a port-and-priority-based policy, use the **priority** parameter.
- To create an ACL-based policy, use the **access-group** parameter.

The **priority q0 | q1 | q2 | q3** parameter specifies the hardware forwarding queue to which the policy applies. Use this parameter only if you are configuring a port-and-priority-based policy. The device prioritizes the queues from **q0** (normal priority) to **q3** (highest priority).

The **access-group** <acl-id> parameter specifies an IP ACL. Use this parameter only if you are configuring an ACL-based policy.

The <average-rate> parameter specifies the maximum number of bits per second (bps) you want the device to allow on the port. You can specify a value in the following ranges:

- Inbound rate limiting on 10/100 Ethernet: 256512 – 100000000 bps.
- Inbound rate limiting on Gigabit Ethernet: 1025792 – 1000000000 bps.
- Outbound rate limiting on 10/100 Ethernet: 1041910 – 100000000 bps.
- Outbound rate limiting on Gigabit Ethernet: 20833792 – 1000000000 bps.

NOTE: The software adjusts the Average Rate you enter so that the calculation of credits does not result in a remainder of a partial Credit. The CLI displays the adjusted rate. You also can display a table of the adjusted rate values. See “Displaying Adjusted Average Rates” on page 9-13.

Layer 2 ACL-Based Rate Limiting

NOTE: This feature is applicable on Foundry devices running Service Provider IronWare software release 09.1.00 or later and Enterprise software release 08.0.00 and later.

JetCore Layer 2 ACL-based rate limiting enables the Foundry device to rate limit incoming traffic in hardware, without CPU intervention. Rate limiting in hardware enables the device to manage bandwidth at line-rate speed.

This feature is an extension to the existing IP ACL-based rate limiting on JetCore devices. Whereas the existing feature provides the facility to limit the rate for IP traffic that matches the permit conditions in standard or extended IP ACLs; the new feature enables you to limit traffic rates using the Layer 2 parameters defined in the associated JetCore Layer 2 ACL table.

In general, Layer 2 ACL-based rate limiting works along the same lines as the JetCore hardware-based rate limiting feature. All the rules and regulations that apply to JetCore rate limiting also apply to this feature.

Configuration Rules and Notes

- Layer 2 ACL-based rate limiting applies only to inbound traffic. You cannot use it to rate limit outgoing traffic.
- You can apply Layer 2 ACL-based rate limiting on a physical port. You cannot apply it to a virtual interface or a trunk port.
- You cannot use Layer 2 ACL-based filtering and Layer 2 ACL-based rate limiting on the same port. You can, however, configure one port on the device to use Layer 2 ACL-based filtering and another port on the same device to use Layer 2 ACL-based rate limiting.
- You cannot use the existing ACL-based rate limiting and Layer 2 ACL-based rate limiting on the same port. However, you can configure one port on the device to use ACL-based rate limiting and another port on the same device to use Layer 2 ACL-based rate limiting.
- You cannot use IP ACLs and Layer 2 ACL-based rate limiting on the same port. You can, however, configure one port on the device to use IP ACLs and another port on the same device to use Layer 2 ACL-based rate limiting.
- By default, when Layer 2 ACL-based rate limiting is enabled on the port, the device rate limits the traffic in hardware. However, when other CPU-based features, such as NetFlow and MAC filters are also enabled on the port, traffic is sent to the CPU for processing and is not subject to rate limiting.

Configuring Layer 2 ACL-Based Rate Limiting

To configure Layer 2 ACL-based rate limiting, perform the following steps:

1. Configure a Layer 2 ACL table with all the necessary clauses. See “Layer 2 ACLs” on page 5-1.
2. Configure a rate limit policy on a physical port using the Layer 2 ACL table ID and the desired average rate. Enter a command such as the following:

```
NetIron 1500(config)# int e 4/25
NetIron 1500(config-if-e1000-4/25)# rate-limit in access-group 400 10000000
```

Syntax: Syntax: [no] rate-limit in access-group <acl-id> <average-rate>

The <acl-id> for Layer 2 ACLs can range from 400 to 499.

The <average-rate> is the maximum number of bits the policy allows during one second.

Editing a Layer 2 ACL Table

You can make changes to the Layer 2 ACL table definitions without unbinding and rebinding the rate limit policy. For example, you can add a new clause to the ACL table, delete a clause from the table, or delete the ACL table that is used by a rate limit policy. See “Layer 2 ACLs” on page 5-1.

Excluding Control Traffic from Rate Limiting

By default, the Layer 2 ACL rate limiting feature does not implement control traffic exemption for rate limiting. You can do so by configuring a respective Layer 2 ACL table and appropriate rate limit policy. For example, to prevent the Foundry device from rate limiting OSPF control traffic, define an ACL table with the following clauses and use it to define the rate limit policy.

```
NetIron 1500(config)# access-list 400 deny any 0000.5E00.0005 ffff.ffff.ffff any
NetIron 1500(config)# access-list 400 deny any 0000.5E00.0006 ffff.ffff.ffff any
NetIron 1500(config)# access-list 400 permit any any any
NetIron 1500(config-if-e1000-4/25)# rate-limit in access-group 400 10000000
```

This configuration defines deny clauses for OSPF control packets which prevent them from being rate limited. The last clause rate limits all other traffic to 10 Mbps. Note that the deny clause will allow traffic to be forwarded without being rate limited, only if the strict ACL mode is not turned on.

Strict ACL Mode

The Layer 2 ACL rate limiting feature includes support for strict ACL mode. By default, Layer 2 ACL clauses with a **deny** action are not subject to rate limiting, and the device forwards all traffic that match these clauses in hardware. You can override this behavior by using strict ACL mode to drop the traffic that matches the deny clauses. For more about strict ACL mode, see the chapter “Enabling Strict TCP or UDP Mode for Flow-Based ACLs” on page 6-55.

Using ACLs for Filtering in Addition to Rate Limiting

When you use the ACL-based mode, the permit and deny conditions in an ACL you use in a rate limiting policy work as follows:

- Permit – The traffic is rate limited according to the other parameters in the rate limiting policy.
- Deny – The traffic is forwarded instead of dropped, by default.

You can configure the device to drop traffic that is denied by the ACL instead of forwarding the traffic, on an individual port basis.

If you use the port-based or port-and-priority-based mode, ACLs forward or drop traffic based on the permit and deny conditions.

NOTE: Once you configure an ACL-based rate limiting policy on a port, you cannot configure a regular (traffic filtering) ACL on the same port. To filter traffic, you must enable the strict ACL option.

To configure the device to drop traffic that is denied by a rate limiting ACL, enter the following command at the configuration level for the port:

```
BigIron(config-if-1/1)# rate-limit strict-acl
```

Syntax: [no] rate-limit strict-acl

Rate Limiting for IP Interface Traffic on JetCore Devices

Beginning with Enterprise software release 08.0.00, you can set a limit for the rate for the traffic, which is directed to IP interfaces on the Foundry device, is sent to the management CPU for processing. This can prevent the management CPU from being overwhelmed when a high rate of traffic is directed to IP interfaces.

When this feature is enabled, the Foundry device rate-limits traffic for which Layer 3 CAM entries have been programmed without next-hop information (zeros). This includes IP traffic, NAT, and local sub-net interface traffic. The rate limit is applied to IP interface traffic on a per DMA basis; that is, the specified rate limit applies to the ports controlled by a given DMA.

To limit the rate for IP interface traffic sent to the CPU, enter a command such as the following:

```
BigIron(config)# intf-proc-bandwidth med
```

Syntax: [no] intf-proc-bandwidth low | med | high

The **low** keyword sets the available processing bandwidth to 1,000,000 bits per second (bps).

The **med** keyword sets the available processing bandwidth to 5,000,000 bps.

The **high** keyword sets the available processing bandwidth to 20,000,000 bps.

The default is 100 Mbps.

In the event that a Denial of Service (DoS) attack sends enough traffic to the IP interfaces that the configured rate limit is reached, it may be difficult to gain access to management functions on the Foundry device through a Telnet or SSH session. To ensure that the Foundry device is reachable even when the rate limit has been applied, you can configure an ACL that explicitly permits traffic from the host to which you are connecting to the Foundry device.

For example, the following commands establish a management VLAN consisting of port 1/1 and an ACL that permits hosts 13.13.13.13 and 10.10.200.18. In the event that the rate limit is reached for IP interface traffic, you should still be able to establish a Telnet or SSH session to the device from one of these hosts.

```
BigIron(config)# vlan 20 by port
BigIron(config-vlan-20)# untagged e 1/1
BigIron(config-vlan-20)# router-interface ve 20
BigIron(config-vlan-20)# exit

BigIron(config)# interface ve 20
BigIron(config-vif-20)# ip address 10.10.210.1 255.255.255.192
BigIron(config-vif-20)# exit

BigIron(config)# access-list 2 permit host 13.13.13.13
BigIron(config)# access-list 2 permit host 10.10.200.18
BigIron(config)# telnet access 2 vlan 20
```

Displaying Rate Limiting Information

You can display the following information:

- The policies that are in effect
- The adjusted Average Rates

Displaying the Policies

To display all the policies on the device, enter the following command at any level of the CLI. This example shows rate limiting polices on a device that is using the port-based rate limiting mode.

```
BigIron(config-if-1/1)# show rate-limit hardware-rate-limit-status
*****************************************
*        Inbound JetCore Rate Limiting      *
*****************************************

Module: 1
 IPC number: 1
  Rate Limit Mode:  Port Based
  Time Interval:    13*0.0192 (ms)
  Credit Size:      32
  Gig Enabled:      Yes

  Port: 1/1, Rate: 3077120(bits/sec), Priority Queue: all, Dir: inbound, ACL: none
  Port: 1/2, Rate: 6153984(bits/sec), Priority Queue: all, Dir: inbound, ACL: none

 IPC number: 2
  Rate Limit Mode:  Port Based
  Time Interval:    13*0.0192 (ms)
  Credit Size:      32
  Gig Enabled:      Yes

  Port: 1/6, Rate: 6153984(bits/sec), Priority Queue: all, Dir: inbound, ACL: none
  Port: 1/2, Rate: 3077120(bits/sec), Priority Queue: all, Dir: inbound, ACL: none

Module: 2
 IPC number: 1
  Rate Limit Mode:  Port Based
  Time Interval:    13*0.0192 (ms)
  Credit Size:      32
  Gig Enabled:      Yes

  Port: 2/2, Rate: 3077120(bits/sec), Priority Queue: all, Dir: inbound, ACL: none
  Port: 2/3, Rate: 3077120(bits/sec), Priority Queue: all, Dir: inbound, ACL: none

 IPC number: 2
  Rate Limit Mode:  Port Based
  Time Interval:    13*0.0192 (ms)
  Credit Size:      32
  Gig Enabled:      Yes

  Port: 2/7, Rate: 6153984(bits/sec), Priority Queue: all, Dir: inbound, ACL: none
  Port: 2/8, Rate: 6153984(bits/sec), Priority Queue: all, Dir: inbound, ACL: none
```



```
*****************************************
*         Outbound JetCore Rate Limiting      *
*****************************************

Module: 1
 IPC number: 1
  Rate Limit Mode:  Port Based
  Time Interval:    32*0.0192 (ms)
  Credit Size:      64
  Gig Enabled:      Yes
  Port: 1/3, Rate: 30000128(bits/sec), Priority Queue: all, Dir: outbound, ACL:
none

 IPC number: 2
  Rate Limit Mode:  Port Based
  Time Interval:    32*0.0192 (ms)
  Credit Size:      64
  Gig Enabled:      Yes

  Port: 1/8, Rate: 60000256(bits/sec), Priority Queue: all, Dir: outbound, ACL:
none

Module: 2
 IPC number: 1
  Rate Limit Mode:  Port Based
  Time Interval:    32*0.0192 (ms)
  Credit Size:      64
  Gig Enabled:      Yes

  Port: 2/2, Rate: 30000128(bits/sec), Priority Queue: all, Dir: outbound, ACL:
none
  Port: 2/3, Rate: 30000128(bits/sec), Priority Queue: all, Dir: outbound, ACL:
none

 IPC number: 2
  Rate Limit Mode:  Port Based
  Time Interval:    32*0.0192 (ms)
  Credit Size:      64
  Gig Enabled:      Yes

  Port: 2/5, Rate: 30000128(bits/sec), Priority Queue: all, Dir: outbound, ACL:
none
```

Syntax: show rate-limit hardware-rate-limit-status

This display shows the following information.

Table 9.5: Rate Limiting Policy Information

This Line...	Displays...
Module	Indicates the forwarding module.

Table 9.5: Rate Limiting Policy Information (Continued)

This Line...	Displays...
IPC number	The IGC or IPC that the rate limiting information is for. Each Gigabit Ethernet module has two IGCs. • IGC 1 manages ports 1 – 4 on the module. • IGC 2 manages ports 5 – 8 on the module. Each 10/100 module has two IPCs: • IPC 1 manages ports 1 – 24 on the module. • IPC 2 manages ports 25 – 48 on the module. On the FastIron 4802, IPC 1 manages ports 1 – 24 and 49. IPC 2 manages ports 25 – 48 and 50.
Rate Limit Mode	The rate limiting mode that is enabled on the device. The mode can be one of the following: • Port Based • Port and Priority Based • L3/L4 Based The L3/L4 Based mode is the same as the ACL-based mode.
Time Interval	The length of each Rate Limiting Interval.
Credit Size	The number of bytes a Credit contains.
Gig Enabled	Whether a rate limiting policy has been configured on a Gigabit port.

Port	List the policies in effect on each port. Each row of information shows the following:
	• Port number
	• Average Rate
	• Hardware forwarding queue
	  • Can be q0, q1, q2, or q3 or "all" for port-and-priority-based policies.
	  • Can be "all" for the other modes.
	• Traffic direction
	• ACL number
	  • "none" for port-based and port-and-priority-based modes.
	  • An ACL number for ACL-based mode.

Displaying Adjusted Average Rates

The CLI automatically adjusts the Average Rate that you enter to ensure that the rate limiting calculation results in a whole number of Credits. You can display the adjusted Average Rates that the CLI will use.

Displaying Adjusted Rates for Inbound Rate Limiting

To display the adjusted rates for a specific range of Average Rates for inbound rate limiting, enter a command such as the following:

```
BigIron# show rate-limit adjusted-rate inbound 2000000 3000000
On 10/100 ports:
Time interval:  640 * 0.0192 ms
Rate 2000000 to 2179487 (bits/sec) will be mapped to 2051328 (bits/sec)
Rate 2179488 to 2435897 (bits/sec) will be mapped to 2307840 (bits/sec)
Rate 2435898 to 2692307 (bits/sec) will be mapped to 2564352 (bits/sec)
Rate 2692308 to 2948717 (bits/sec) will be mapped to 2820608 (bits/sec)
Rate 2948718 to 3000000 (bits/sec) will be mapped to 3077120 (bits/sec)

On Gig ports:
Time interval:  13 * 0.0192 ms
Rate 2000000 to 2564102 (bits/sec) will be mapped to 2051328 (bits/sec)
Rate 2564103 to 3000000 (bits/sec) will be mapped to 3077120 (bits/sec)
```

This example shows the adjusted rates for Average Rates between 2000000 and 3000000 bps. The rates for 10/100 Ethernet ports and Gigabit Ethernet ports are different and are listed separately.

Syntax: show rate-limit adjusted-rate inbound <start-rate> <end-rate>

The **inbound** parameter specifies that you want to display rates for inbound rate limiting. The adjusted rates for inbound rate limiting and outbound rate limiting are not the same. To display rates for outbound rate limiting, use the command in "Displaying Adjusted Rates for Outbound Rate Limiting" below.

The <start-rate> <end-rate> parameter specifies the range of Average Rates for which you want to list the adjusted rates. You can specify a range of up to 10000000 (10 million) bps. For example, you can specify 10000000 to 19999999, but not 10000000 to 20000000.

Displaying Adjusted Rates for Outbound Rate Limiting

To display the adjusted rates for a specific range of Average Rates for outbound rate limiting, enter a command such as the following:

```
BigIron# show rate-limit adjusted-rate outbound gig-port 30000000 40000000

Time interval:  32 * 0.0192 ms
Rate 30000000 to 30416665 (bits/sec) will be mapped to 30000128 (bits/sec)
Rate 30416666 to 31249997 (bits/sec) will be mapped to 30833664 (bits/sec)
Rate 31249998 to 32083330 (bits/sec) will be mapped to 31666688 (bits/sec)
Rate 32083331 to 32916665 (bits/sec) will be mapped to 32500224 (bits/sec)
Rate 32916666 to 33749997 (bits/sec) will be mapped to 33333760 (bits/sec)
Rate 33749998 to 34583330 (bits/sec) will be mapped to 34166784 (bits/sec)
Rate 34583331 to 35416666 (bits/sec) will be mapped to 35000320 (bits/sec)
Rate 35416667 to 36249997 (bits/sec) will be mapped to 35833344 (bits/sec)
Rate 36249998 to 37083330 (bits/sec) will be mapped to 36666880 (bits/sec)
Rate 37083331 to 37916666 (bits/sec) will be mapped to 37500416 (bits/sec)
Rate 37916667 to 38749997 (bits/sec) will be mapped to 38333440 (bits/sec)
Rate 38749998 to 39583330 (bits/sec) will be mapped to 39166976 (bits/sec)
Rate 39583331 to 40000000 (bits/sec) will be mapped to 40000512 (bits/sec)
```

This command shows the adjusted rates between 30000000 and 40000000 bps for outbound rate limiting on a Gigabit Ethernet port.

Syntax: show rate-limit adjusted-rate outbound gig-port | non-gig-port <start-rate> <end-rate>

The **outbound** parameter specifies that you want to display rates for outbound rate limiting. The adjusted rates for inbound rate limiting and outbound rate limiting are not the same. To display rates for inbound rate limiting, use the command in “Displaying Adjusted Rates for Inbound Rate Limiting“ above.

The **gig-port | non-gig-port** parameter specifies the port type. The valid rates differ depending on the port type.

The <start-rate> <end-rate> parameter specifies the range of Average Rates for which you want to list the adjusted rates. You can specify a range of up to 10000000 (10 million) bps. For example, you can specify 10000000 to 19999999, but not 10000000 to 20000000.

Fixed Rate Limiting

This release introduces Fixed Rate Limiting for outbound ports of devices with JetCore modules. This style of rate limiting is in addition to the Fixed Rate Limiting introduced in software release 09.1.00 for inbound ports.

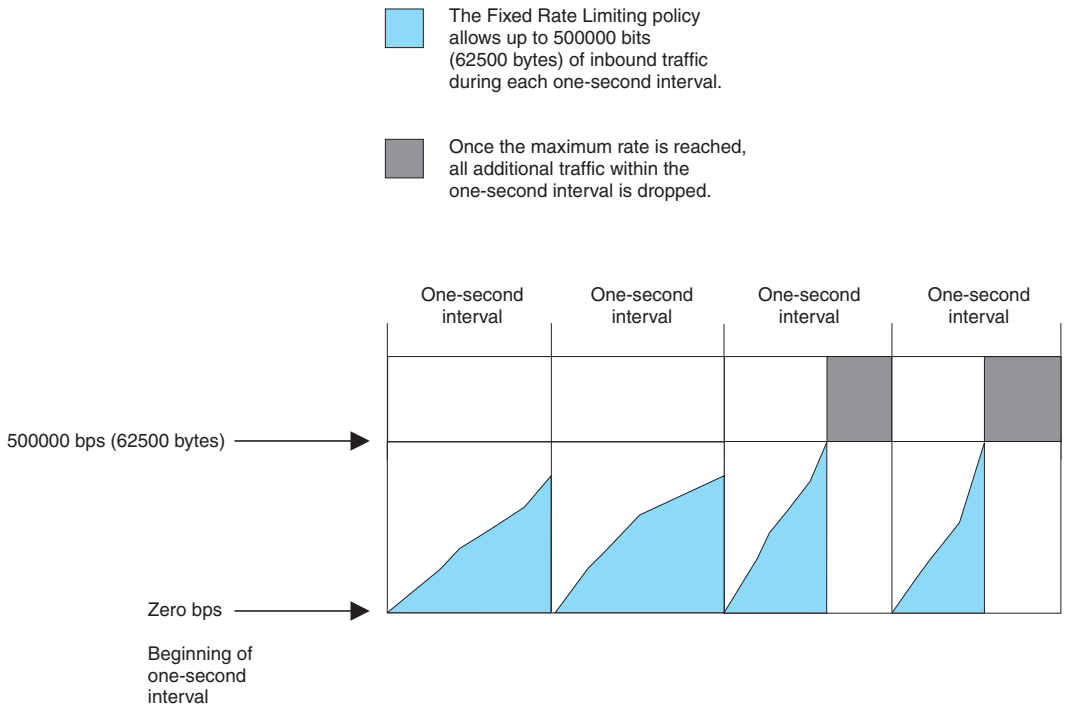
NOTE: Fixed Rate Limiting for inbound and outbound ports are supported on all JetCore modules running software release 09.1.02, except for the following: J-24FX module and all 10-Gigabit Ethernet modules

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 configured the rate, the inbound port drops all further packets during the one-second interval.

Once the one-second interval is complete, the port clears the counter and re-enables traffic.

Figure 9.1 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 9.1 Fixed Rate Limiting



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

To configure a fixed rate limiting policy, enter a command such as the following at the configuration level for a port:

```
NetIron(config-if-1/1)# rate-limit input fixed 100000000
```

This command restricts the number of bytes that port 1/1 receives to 100 Mbits per second. Additional bytes are dropped.

You can also enter the following command:

```
NetIron(config-if-1/1)# rate-limit output fixed 100000000
```

This command restricts the number of bytes that port 1/1 sends to 100 Mbits per second. Additional bytes are dropped.

Syntax: [no] rate-limit input | outbound fixed <rate>

The **input | outbound** parameter specifies whether the rate limit applies to inbound or outbound traffic on the port.

The <rate> parameter specifies the maximum rate for the port. Specify the rate in bits per second. You can specify from 1 up to any number. There is no default.

Use the **no** parameter to remove the inbound or outbound rate limiting policy from the port.

Displaying Fixed Rate Limiting Information

To display configuration information and statistics for Fixed Rate Limiting, enter the following command at any level of the CLI:

```
NetIron(config)# show rate-limit fixed

Total rate-limited interface count: 6.
    Port     Input rate   RX Enforced    Output rate   TX Enforced
     1/1         500000             3
     2/1                                    1234567           100
     2/2                                    2222222             3
     2/3                                    1234567            15
     2/4                                    1238888            12
     2/5                                    1238888             7
```

Syntax: show rate-limit fixed

This display shows the following information.

Table 9.6: 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.
Input rate	The maximum rate allowed for inbound traffic. The rate is measured in bits per second (bps).

Table 9.6: CLI Display of Fixed Rate Limiting Information (Continued)

This Field...	Displays...
RX Enforced	The number of one-second intervals in which the fixed rate limiting policy has dropped traffic received on the port.
Output rate	The maximum rate allowed for outbound traffic. The rate is measured in bps.
TX Enforced	The number of one-second intervals in which the fixed rate limiting policy has dropped traffic queued to be sent on the port.

Once the command is entered and the statistics are displayed, the statistics are cleared for the ports in the report.

NOTE: Packets corresponding to the excess traffic are dropped due to the application of outbound fixed rate limiting. These packets are accounted for in the WriteDrops counter of the **hw bc** diagnostics command. It is normal to see the WriteDrops counter increment when the outbound fixed rate limiting feature is in use. This diagnostics display change is specific only to the Jetcore fixed rate limiting on outbound ports.

Clearing Rate Limiting Statistics

To reset the RX Enforced and TX Enforced values for a port, enter the following command:

```
NetIron(config)# clear statistics rate-counters
```

Syntax: clear statistics rate-counters

Entering a **show rate-limit fixed** command after clearing the rate limiting statistics, shows that the RX Enforced and TX Enforced have been reset. For example:

```
NetIron(config)# show rate-limit fixed

Total rate-limited interface count: 6.
    Port     Input rate   RX Enforced    Output rate   TX Enforced
     1/1         500000             0
     2/1                                    1234567             3
     2/2                                    2222222             0
     2/3                                    1234567             2
     2/4                                    1238888             1
     2/5                                    1238888             0
```


Chapter 10
Configuring Rate Limiting on NetIron IMR 640

Rate Limiting on NetIron IMR 640 in Release 02.0.02

The NetIron IMR 640 provides line-rate rate limiting in hardware on inbound ports and outbound ports.

You can configure the NetIron IMR 640 to use one of the following modes of rate limiting policies:

- Port-based – Limits the rate on an individual physical port to a specified rate. Only one inbound and one outbound port-based rate limiting policy can be applied to a port. (Refer to “Configuring Port-Based Rate Limiting For Inbound and Outbound Ports” on page 10-4.) These policies can be applied to inbound and outbound traffic.
- Port-and-priority-based – Limits the rate on an individual hardware forwarding queue on an individual physical port. Only one port-and-priority-based rate limiting policy can be specified per priority queue for a port. (Refer to “Configuring a Port and Priority-Based Rate Limiting Policy for Inbound and Outbound Ports” on page 10-5.) These policies can be applied to inbound and outbound traffic.
- Port-and-VLAN-based – Limits the rate of packets tagged with a specific VLAN on an individual physical port. Only one rate can be specified for each VLAN. (Refer to “Configuring a Port-and-VLAN-Based Rate Limiting Policy” on page 10-5.) Up to 1024 VLAN-based policies can be configured for a port under normal conditions or 4096 policies if priority-based rate limiting is disabled as described in “Configuring for No Priority-Based Rate Limiting” on page 10-8. These policies can be applied to inbound and outbound traffic.
- VLAN group based – Limits the traffic for a group of VLANs. Members of a VLAN group share the specified bandwidth defined in the rate limiting policy that has been applied to that group. (Refer to “Configuring a Port-and-VLAN Group-Based Rate Limiting Policy” on page 10-6.) Up to 1024 VLAN Group-based policies can be configured for a port under normal conditions or 4096 policies if priority-based rate limiting is disabled as described in “Configuring for No Priority-Based Rate Limiting” on page 10-8. These policies can only be applied to inbound traffic.
- Port-and-ACL-based – Limits the rate of IP traffic on an individual physical port that matches the permit conditions in IP Access Control Lists (ACLs). **Layer-2 ACL-based rate-limiting is supported**. You can use standard or extended IP ACLs. Standard IP ACLs match traffic based on source IP address information. Extended ACLs match traffic based on source and destination IP address and IP protocol information. Extended ACLs for TCP and UDP also match on source and destination TCP or UDP addresses. and protocol information. These policies can be applied to inbound and outbound traffic. Up to 1024 Port-and-ACL-based policies can be configured for a port under normal conditions or 4096 policies if priority-based rate limiting is disabled as described in “Configuring for No Priority-Based Rate Limiting” on page 10-8.

This release of Multi-Service IronWare supports applying rate limiting parameters directly to a port or creating a policy map to define a set of rate limiting parameters and then applying that policy map to one or more ports. In addition, the rate limiting parameters available from each of these options are different. The parameters used when applying rate limiting parameters directly to a port reflect the Multi-Service IronWare features that were

available before this release. These parameters and the information required to use them are described in “Applying Rate Limiting Parameters Directly to a Port” on page 10-2.

The parameters used when applying rate limiting through use of a policy map reflect the rate limiting features that have been added with this release. These parameters and the information required to use them are described in “Applying Rate Limiting Parameters Using a Policy Map” on page 10-3.

Applying Rate Limiting Parameters Directly to a Port

When applying a rate limiting policy directly to a port, there are specific parameters that are applied to implement the policy that are different than those used when using a policy map. This method was previously employed in the NetIron IMR 640 router and the NetIron IMR 640 supports this mode in addition to the new mode supported using a policy map. Using this method, a rate limiting policy specifies two parameters: average rate and maximum burst. These parameters are used to configure credits and credit totals.

Average Rate

The *Average Rate* is the maximum number of bits a port is allowed to receive during a one-second interval. The rate of the traffic that matches the rate limiting policy will not exceed the average rate.

The Average Rate represents a percentage of an interface's line rate (bandwidth), expressed in bits per second (bps). It cannot be smaller than 53,328 bits per second (bps) and it cannot be larger than the port’s line rate.

Average Rate must be entered in multiples of 53,328 bps. If you enter a number that is not a multiple of 53,328, the software adjusts the rate down to the lowest multiple of the number so that the calculation of credits does not result in a remainder of a partial Credit. For example, if you enter 60,000 bps, the value will be adjusted to 53,328 bps. The adjusted rate is sometimes called the *adjusted average rate*.

Maximum Burst

Maximum burst provides a higher than average rate to traffic that meet the rate limiting criteria. Traffic will be allowed to pass through the port for a short period of time. The unused bandwidth can be accumulated up to a maximum of “maximum burst” value.

Credits and Credit Total

Each rate limiting policy is assigned a class. A *class* uses the average rate and maximum allowed burst in the rate limit policy to calculate credits and credit totals.

Credit size is measured in bytes. A credit is a forwarding allowance for a rate-limited port, and is the smallest number of bytes that can be allowed during a rate limiting interval. Minimum credit size can be 1 byte.

During a rate limiting interval, a port can send or receive only as many bytes as the port has Credits for. For example, if an inbound rate limiting policy results in a port receiving two credits per rate limiting interval, the port can send or receive a maximum of 2 bytes of data during that interval.

The credit size is calculated using the following algorithm:

Credit = (Average rate in bits per second)/(8*6666)

One second is divided into 10,000 intervals. In each interval, the number of bytes equal to the credit size is added to the running total of the class. The running total of a class represents the number of bytes that can be allowed to pass through without being subject to rate limiting.

The second parameter is the maximum *credit total*, which is also measured in bytes. The maximum credit total is calculated using the following algorithm.

Maximum credit total = (Maximum burst in bits)/8

The running total can never exceed the maximum credit total. When packets arrive at the port, a class is assigned to the packet based on the rate limiting policies. If the running total of the class is less than the size of the packet, then the packet is dropped. Otherwise, the size of the packet is subtracted from the running total and the packet is forwarded. If there is no traffic that matches rate limiting criteria, then the running total can grow up to the maximum credit total.

Applying Rate Limiting Parameters Using a Policy Map

When using the rate limiting policies available from previous versions, the policy parameters are provided explicitly for each port during port configuration. In this version, the policies must be defined using a policy map. The policy map configuration ties a policy name to a set of rate limiting policies. The policy name is then applied to the port or ports that you want to rate limit using the defined policy. This allows you to set a policy in a single location the affects multiple ports and to make changes to that policy. Configuration of a policy map is described in “Configuring a Policy Map” on page 10-4.

Within the policy map configuration, the parameters used to define rate limiting have been changed. When configuring rate limiting within a policy map, these new parameters apply. With this release, rate limiting policy determines the rate of inbound or outbound traffic (in bits per second or bps) that is allowed per port. This traffic is initially rate limited by a Committed Information Rate (CIR) bucket. Traffic that is not accommodated in the CIR bucket is then subject to the Excess Information Rate (EIR) bucket.

The CIR Bucket

The CIR rate limiting bucket is defined by two separate parameters: the CIR rate, and the Committed Burst Size (CBS) rate. The CIR rate is the maximum number of bits a port is allowed to receive or send during a one-second interval. The rate of the traffic that matches the rate limiting policy can not exceed the CIR rate. The CIR rate represents a portion of an interface's line rate (bandwidth), expressed in bits per second (bps) and it cannot be larger than the port’s line rate. CIR-defined traffic that does not use the CIR rate available to it accumulates credits that it can use later in circumstances where it temporarily exceeds the CIR rate.

When traffic exceeds the bandwidth that has been reserved for it by the CIR rate defined in its policy, it becomes subject to the CBS rate. The CBS rate provides a rate higher than the CIR rate to traffic that exceeded its CIR rate. The bandwidth in the CBS rate is accumulated during periods of time when traffic that has been defined by a policy does not use the full CIR rate available to it. Traffic is allowed to pass through the port for a short period of time at the CBS rate.

When inbound or outbound traffic exceeds the bandwidth available for the defined CIR and CBS rates, it is either dropped, or made subject to the conditions set in it EIR bucket.

The EIR Bucket

The EIR bucket provides an option for traffic that has exceeded the conditions set by policy for the CIR bucket. In the EIR bucket, there are two parameters that define the traffic that is available: the Excess Information Rate (EIR) and the Excess Burst Size (EBS) rate. The EIR and EBS operate exactly like the CIR and CBS except that they only act upon traffic that has been passed to the EIR bucket because it could not be accommodated by the CIR bucket. Like the CIR, the EIR provides an initial bandwidth allocation to accommodate inbound and outbound traffic. If the bandwidth provided by the EIR is insufficient to accommodate the excess traffic, the defined EBS rate provides for burst traffic. Like the CBS, the bandwidth available for burst traffic from the EBS is subject to the amount of bandwidth that is accumulated during periods of time when traffic that has been allocated by the EIR policy isn’t used.

In addition, to providing additional bandwidth for traffic that exceeds that available for the CIR bucket, traffic rate limited by the EIR bucket can have its excess priority and excess dscp values changed. Using this option, priority parameters are set following the EBS value that change the priority of traffic that is being rate limited using the EIR bucket.

Configuration Considerations

- Only one type of rate limiting policy can be applied on a physical port. For example, you cannot apply port-and-ACL-based and port-based rate limiting policies on the same port.
- When a port-and-VLAN-based rate limiting policy is applied to a port, all the ports controlled by the same packet processor are rate limited for that VLAN. You cannot apply a port-and-VLAN-based rate limiting policy on another port of the same packet processor for the same VLAN ID.
- Port-and-VLAN-based rate limiting can limit only tagged packets that match the VLAN ID specified in the policy. Untagged packets are not subject to rate limiting.
- The maximum burst in a rate limit policy cannot be less than the average rate and cannot be more than the port’s line rate.

- Control packets are not subject to rate limiting.

Configuring Rate Limiting on NetIron IMR 640 Devices

The following sections show examples of how to configure each rate limiting policy type.

Configuring a Policy Map

To configure a policy map:

```
NI IMR640 Router(config)# policy-map map1 cir 1000000 cbs 2000000 eir 1000000 ebs
2000000 excess-priority 1 excess-dscp 30
```

The command configures the rate limiting policy map map1 to limit CIR rate to 1000000 the CBS rate to 2000000, the EIR rate to 1000000 and the EBS to 2000000. In addition, traffic that exceeds the bandwidth available in the CIR bucket will have its packets priority queue set to 1 and its DSCP set to 30. This command only creates a policy, it must be applied to one or more ports to be operational.

Syntax: [no] policy-map <map-name> cir <cir-rate> cbs <cbs-rate> [eir <eir-rate> ebs <ebs-rate> excess-priority <priority-num> [excess-dscp <dscp-num>]]

The map-name variable is the name you will use to reference the policy map in rate limit command. It can be a character string up to 64 characters long.

The **cir** parameter defines the value of the Committed Information Rate (CIR) as the rate defined in the <cir-rate> variable. Acceptable values are: 0 - 10000000000 bps in increments of 53,328 bps.

The **cbs** parameter defines the value of the Committed Burst Size (CBS) as the rate defined in the <cbs-rate> variable. Acceptable values are: 1250 - 1250000000 bytes in increments of 1 byte.

The **eir** parameter defines the value of the Excess Information Rate (EIR) as the rate defined in the <eir-rate> variable. Acceptable values are: 0 - 10000000000 bps in increments of 53,328 bps.

The **ebs** parameter defines the value of the Excess Burst Size (EBS) as the rate defined in the <ebs-rate> variable. Acceptable values are: 1250 - 1250000000 bytes in increments of 1 byte.

The **excess-priority** parameter specifies that traffic whose bandwidth requirements exceeds what is available in the CIR bucket and is sent to the EIR bucket will have its packets priority queue set to the value set in the <priority-num> variable. Acceptable values for the <priority-num> are 0-3.

The **excess-dscp** parameter specifies that traffic whose bandwidth requirements exceeds what is available in the CIR bucket and is sent to the EIR bucket will have its packets DSCP priority set to the value set in the <dscp-num> variable. Acceptable values for the <dscp-num> are 0-63.

Configuring Port-Based Rate Limiting For Inbound and Outbound Ports

Port-based rate limiting limits the rate on an individual inbound or outbound physical port to a specified rate.

To configure port-based rate limiting policy for outbound ports, enter commands such as the following at the interface level:

```
NetIron IMR 640(config)# interface ethernet 1/1
NetIron IMR 640(config-if-1/1)# rate-limit out 500000000 750000000
```

The commands configure a rate limiting policy for outbound traffic on port 1/1. The policy limits the average rate of all outbound traffic to 500 Mbps with a maximum burst size of 750 Mbps.

To configure port based rate limiting policy through a policy map:

```
NI IMR640 Router(config)# interface ethernet 1/1
NI IMR640 Router(config-if-1/1)# rate-limit input policy-map map1
```

The commands configure a rate limiting policy for inbound traffic on port 1/1. The policy references the policy map map1 for rate limiting policy parameters.

The complete syntax for configuring a port-based rate limiting policy is:

Syntax: [no] rate-limit [in | out] [<average-rate> <maximum-burst> | policy-map <map-name>]

The **input** parameter applies the policy to traffic on inbound ports.

The **output** parameter applies the policy to traffic on outbound ports.

Only one inbound and one outbound port-based rate limiting policy can be applied to a port.

The <average-rate> parameter specifies the maximum rate allowed on a port during a one-second interval. The software automatically adjusts the number you enter to the lower multiple of 53,328 bps. Refer to the section "Average Rate" on page 10-2 for more details. This command is only used when configuring rate limiting directly to a port as described in "Applying Rate Limiting Parameters Directly to a Port" on page 10-2.

The <maximum-burst> parameter specifies the extra bits above the average-rate that traffic can have. Refer to the section "Maximum Burst" on page 10-2 for more details. This command is only used when configuring rate limiting directly to a port as described in "Applying Rate Limiting Parameters Directly to a Port" on page 10-2.

The **policy-map** parameter specifies the policy map named in the <policy-map> variable to be used to provide parameters for rate limiting the port and VLAN specified. This command is only used when configuring rate limiting to a port using a policy map as described in "Applying Rate Limiting Parameters Using a Policy Map" on page 10-3.

Configuring a Port and Priority-Based Rate Limiting Policy for Inbound and Outbound Ports

To configure port based rate limiting policy directly:

```
NI IMR640 Router(config)# interface ethernet 1/1
NI IMR640 Router(config-if-1/1)# rate-limit input priority 2 500000000 750000000
Average rate is adjusted to 499639656 bits per second
```

The commands configure a rate limiting policy for inbound traffic on port 1/1. The policy limits the average rate of all inbound traffic to 500 Mbps with a maximum burst size of 750 Mbps for packets with their priority queue set to 2.

Syntax: [no] rate-limit [input | output] priority <queue-num> [<average-rate> <maximum-burst> | policy-map <map-name>]

The **input** parameter applies the policy to traffic on inbound ports.

The **output** parameter applies the policy to traffic on outbound ports.

Only one port-based rate limiting policy can be applied to a port.

The **priority** parameter specifies an 802.1p value in the <queue-num> variable that is used to identify packets that will be rate limited by this command.

The <average-rate> parameter specifies the maximum rate allowed on a port during a one-second interval. The software automatically adjusts the number you enter to the nearest multiple of 515,624 bps. Refer to the section "Average Rate" on page 10-2 for more details. This command is only used when configuring rate limiting directly to a port as described in "Applying Rate Limiting Parameters Directly to a Port" on page 10-2.

The <maximum-burst> parameter specifies the extra Mbits above the average-rate that traffic can have. Refer to the section "Maximum Burst" on page 10-2 for more details. This command is only used when configuring rate limiting directly to a port as described in "Applying Rate Limiting Parameters Directly to a Port" on page 10-2.

The **policy-map** parameter specifies the policy map named in <map-name>. It is only used when configuring rate limiting to a port using a policy map as described in "Applying Rate Limiting Parameters Using a Policy Map" on page 10-3.

Configuring a Port-and-VLAN-Based Rate Limiting Policy

To configure a port-and-VLAN based rate limiting policy, enter commands such as the following:

```
NI IMR640 Router(config)# interface ethernet 1/1
```



```
NI IMR640 Router(config-if-1/1)# rate-limit input vlan 10 500000000 750000000

NI IMR640 Router(config)# interface ethernet 1/2
NI IMR640 Router(config-if-1/2)# rate-limit output vlan 20 policy-map map1
```

These commands configure two rate limiting policies that limit the average rate of all inbound traffic on port 1/1 with VLAN tag 10 and all outbound traffic on port 1/2 VLAN tag 20. The first policy limits packets with VLAN tag 10 to an average rate of 500 Mbps with a maximum burst size of 750 Mbits on port 1/1. The second policy limits packets with VLAN tag 20 to values defined in policy map map1. Tagged packets belonging to VLANs other than 10 and 20 and untagged packets are not subject to rate limiting on these ports.

Syntax: [no] rate-limit [input | output] [priority <queue-num>] vlan-id <vlan-num> [<average-rate> <maximum-burst> | policy-map <map-name>]

The **input** parameter applies the policy to traffic on inbound ports.

The **output** parameter applies the policy to traffic on outbound ports.

The **priority** parameter specifies an 802.1p value in the <queue-num> variable that is used to identify packets that will be rate limited by this command. This parameter is optional.

The vlan-id <vlan-number> parameter species the VLAN ID to which the policy applies. You can specify up to 10 port-and-VLAN-based rate limit policies on a port.

The <average-rate> parameter specifies the maximum rate allowed on a port during a one-second interval. The software automatically adjusts the number you enter to the nearest multiple of 53,328 bps. Refer to the section “Average Rate” on page 10-2 for more details. This command is only used when configuring rate limiting directly to a port as described in “Applying Rate Limiting Parameters Directly to a Port” on page 10-2.

The <maximum-burst> parameter specifies the extra Mbits above the average-rate that traffic can have. Refer to the section “Maximum Burst” on page 10-2 for more details. This command is only used when configuring rate limiting directly to a port as described in “Applying Rate Limiting Parameters Directly to a Port” on page 10-2.

The **policy-map** parameter specifies the policy map named in the <policy-map> variable to be used to provide parameters for rate limiting the port and VLAN specified. This command is only used when configuring rate limiting to a port using a policy map as described in “Applying Rate Limiting Parameters Using a Policy Map” on page 10-3.

Configuring a Port-and-VLAN Group-Based Rate Limiting Policy

A rate limiting policy can be applied to a VLAN group. VLANs that are members of a VLAN group share the specified bandwidth defined in the rate limiting policy applied to that group.

To configure a rate limiting policy for a VLAN group, do the following:

1. Define the VLANs that you want to place in a rate limiting VLAN group.
2. Define a rate limiting VLAN group. This VLAN group is specific to the rate limiting feature. Enter commands such as the following:

   ```
   NI IMR640 Router(config)# rl-vlan-group 10
   NI IMR640 Router(config-vlan-rate-group)# vlan 3 5 to 7 10
   ```

   The commands assign VLANs 3, 5,6, 7, and 10 to rate limiting VLAN group 10.

   Syntax: [no] rl-vlan-group <vlan-group-number>

   Syntax: [no] vlan <vlan-number> [to <vlan-number]

   The **rl-vlan-group** command takes you to the VLAN group rate limiting level. Enter the ID of the VLAN group that you want to create or update by entering a value for <vlan-group-number>.

   Use the vlan command to assign or remove VLANs to the rate limiting VLAN group. You can enter the individual VLAN IDs or a range of VLAN IDs.
3. Create a policy for the VLAN group and apply it to the interface you want. Enter commands such as the following:


```
NI IMR640 Router(config)# interface ethernet 1/1
NI IMR640 Router(config-if-1/1)# rate-limit input group 10 500000000 750000000
```

These commands configure a rate limiting policy that limits the average rate of all inbound traffic on port 1/1 from VLAN group 10. This policy limits packets from VILAN group 10 to an average rate of 500 Mbps with a maximum burst size of 750 Mbits on port 1/1.

Port and VLAN Group based rate limiting is only available for inbound ports.

Syntax: [no] rate-limit input group <vlan-group-id> [priority <queue-num>] [<average-rate> <maximum-burst> | policy-map <map-name>]

The <vlan-group-id> parameter specifies the VLAN GroupID to which the policy applies.

The **priority** parameter specifies an 802.1p value in the <queue-num> variable that is used to identify packets that will be rate limited by this command. This parameter is optional.

The <average-rate> parameter specifies the maximum rate allowed on a port during a one-second interval. The software automatically adjusts the number you enter to the nearest multiple of 53,328 bps. Refer to the section "Average Rate" on page 10-2 for more details. This command is only used when configuring rate limiting directly to a port as described in "Applying Rate Limiting Parameters Directly to a Port" on page 10-2.

The <maximum-burst> parameter specifies the extra Mbits above the average-rate that traffic can have. Refer to the section "Maximum Burst" on page 10-2 for more details. This command is only used when configuring rate limiting directly to a port as described in "Applying Rate Limiting Parameters Directly to a Port" on page 10-2.

The **policy-map** parameter specifies the policy map named in <map-name>. It is only used when configuring rate limiting to a port using a policy map as described in "Applying Rate Limiting Parameters Using a Policy Map" on page 10-3.

Configuring a Port-and-ACL-Based Rate Limiting Policy

You can use standard or extended IP ACLs for port-and-ACL-based rate limiting.

- Standard IP ACLs match traffic based on source IP address information.
- Extended ACLs match traffic based on source and destination IP addresses and IP protocol information. Extended ACLs for TCP and UDP protocol must also match on source and destination IP addresses and TCP or UDP protocol information.
- You can apply an ACL ID to a port-and-ACL-based rate limiting policy even before you define the ACL. The rate limiting policy does not take effect until the ACL is defined.
- It is not necessary to remove an ACL from a port-and-ACL-based rate limiting policy before deleting the ACL.
- Layer-2 ACL rate limiting is supported.

Port-and-ACL-based rate limiting is supported for traffic on inbound and outbound ports. To configure port-and-ACL-based rate limiting policies, enter commands such as the following:

```
NI IMR640 Router(config)#access-list 50 permit host 1.1.1.2
NI IMR640 Router(config)#access-list 50 deny host 1.1.1.3
NI IMR640 Router(config)#access-list 60 permit host 2.2.2.3
NI IMR640 Router(config-if-1/1)# rate-limit input access-group 50 priority q1
500000000 750000000
NI IMR640 Router(config-if-1/1)# rate-limit input access-group 60 100000000
200000000
```

These commands first configure access-list groups that contain the ACLs that will be used in the rate limiting policy. Use the **permit** condition for traffic that will be rate limited. Traffic that match the **deny** condition are not subject to rate limiting.

Next, the commands configure two rate limiting policies on port 1/1. The policies limit the average rate of all inbound IP traffic that match the permit rules of ACLs 50 and 60. The first policy limits the rate of all permitted IP traffic with a priority queue value of q1 from host 1.1.1.2 to an average rate of 500 Mbps with a maximum burst

size of 750 Mbits. Rate of all traffic from host 1.1.1.3 is not subject to rate limiting since it is denied by ACL 50; it is merely forwarded on the port.

The second policy limits the rate of all IP traffic from host 2.2.2.3 to an average rate of 100 Mbps with a maximum burst size of 200 Mbits.

All IP traffic that does not match ACLs 50 and 60 are not subject to rate limiting.

Syntax: [no] rate-limit [input | output] [vrf <vrf-name>] access-group <group-number> [priority <queue-num>] [<average-rate> <maximum-burst> | policy-map <map-name>]

The **input** parameter applies the policy to traffic on inbound ports.

The **output** parameter applies the policy to traffic on outbound ports.

The **VRF** parameter specifies that the access-group will only apply to traffic within the VRF whose name is specified in the <vrf-name> variable. This feature is only supported on inbound traffic with Layer-3 ACLs.

The **access-group**, group-number> parameter specifies the group number to which the ACLs used in the policy belong.

NOTE: An ACL must exist in the configuration before it can take effect in a rate limiting policy.

You can specify up to 10 ACL-based rate limiting policies on a port.

The **priority** parameter specifies a priority queue value in the <queue-num> variable that is used to identify packets that will be rate limited by this command. The possible values for this parameter are: q0, q1, q2, or q3. Multiple queues can be specified. This parameter is optional.

The <average-rate> parameter specifies the maximum rate allowed on a port during a one-second interval. The software automatically adjusts the number you enter to the nearest multiple of 515,624 bps. Refer to the section “Average Rate” on page 10-2 for more details. This command is only used when configuring rate limiting directly to a port as described in “Applying Rate Limiting Parameters Directly to a Port” on page 10-2

The <maximum-burst> parameter specifies the extra Mbits above the average-rate that traffic can have. Refer to the section “Maximum Burst” on page 10-2 for more details. This command is only used when configuring rate limiting directly to a port as described in “Applying Rate Limiting Parameters Directly to a Port” on page 10-2

The **policy-map** parameter specifies the policy map named in <map-name>. It is only used when configuring rate limiting to a port using a policy map as described in “Applying Rate Limiting Parameters Using a Policy Map” on page 10-3.

Configuring for No Priority-Based Rate Limiting

By default, up to 1024 different rate limiting policies can be applied to a single 10 GB Ethernet port. This combined with the 4 priorities utilizes 4096 rate limiting classes. You can configure a system-wide policy so that up to 4096 individual rate limiting policies can be applied to a single 10 GB Ethernet port.

To configure a NetIron IMR 640 to not allow priority-based rate limiting, enter commands such as the following at the interface level:

```
NI IMR640 Router(config)# qos-policy
NI IMR640 Router(qos-policy)# no rate-limit internal-priority-based
```

Syntax: [no] rate-limit internal-priority-based

If this command is implemented, the number of different rate limiting policies that can be applied to a single port is increased from 1024 to 4096.

Configuring Egress Priority Merging

In some cases, policies such as ACL and rate limiting may reduce the 802.1p priority status of packets as they leave the router. One such case could be packets that exceed the configured CIR. If you want to preserve the incoming 802.1p priority, enter commands such as the following at the interface level:

```
NI IMR640 Router(config)# qos-policy
```



```
NI IMR640 Router(qos-policy)# merge-egress-priorities
```

Syntax: merge-egress-priorities

Displaying Rate Limiting Policies

Use one of the following commands to view the rate limiting policies that have been configured:

- **show rate-limit counters** – Displays accounting information for rate limit usage.
- **show rate-limit group** – Displays the VLANs that are in the specified group.
- **show rate-limit** – Displays rate limiting policies implemented per interface.
- **show policy-map** – Displays rate limiting policies implemented in the configured policy maps.

Displaying Accounting Information for Rate Limit Usage

To display accounting information for rate limit usage, enter the following command:'

```
NI IMR640 Router# show rate-limit counters
```

Syntax: show rate-limit counters [interface slot/port]

The **interface slot/port** option allows you to get accounting information for a specified interface only.

Output such as the following will display

```
NI IMR640 Router# show rate-limit counters
interface e 1/1
 rate-limit input 959904 2000000
  Fwd:        10000                  Drop:  1000 bytes
  Re-mark:    0                      Total: 11000 bytes
 rate-limit output 2986368 2000000
  Fwd:        20000                  Drop:  2340 bytes
  Re-mark:    0                      Total: 22340 bytes
interface e 1/2
 rate-limit input vlan-id 3 policy-map pmap1
  Fwd:        10200                  Drop:  2350 bytes
  Re-mark:    100                    Total: 12450 bytes
 rate-limit input vlan-id 5 priority q2 policy-map pmap2
  Fwd:        10000                  Drop:  500 bytes
  Re-mark:    700                    Total: 11200 bytes
```

This display shows the following information.

Table 10.1: Rate Limit Counters Parameters

This Field...	Displays...
Interface	The interface that rate limit accounting information is being displayed for.
rate-limit input	A rate limit configuration that defines rate limit policy for inbound traffic on the defined interface.
rate-limit output	A rate limit configuration that defines rate limit policy for outbound traffic on the defined interface.

Table 10.1: Rate Limit Counters Parameters (Continued)

This Field...	Displays...
Fwd	The traffic in bytes that has been forwarded from this interface as a result of this rate limit policy since the router was started up or the counter has been reset.
Drop	The traffic in bytes that has been dropped from this interface as a result of the defined rate limit policy since the router was started up or the counter has been reset.
Re-mark	The number of packets whose priority have been remarked as a result of exceeding the bandwidth available in the CIR bucket for this rate limit policy.
Total	The total traffic in bytes that has been carried on this interface for the defined rate limit policy since the router was started up or the counter has been reset.

Resetting the Rate Limit Counters

You can reset all of the rate limit counters using the following command:

```
NI IMR640 Router# clear rate-limit counters
```

Syntax: clear rate-limit counters [interface]

The **interface** variable specifies an interface that you want to clear the rate limit counters for. If you do not specify an interface, all rate limit counters on the router will be reset.

Displaying Information about rate limit VLAN groups

To display information about rate limit VLAN groups, enter the following command:'

```
NI IMR640 Router# show rate-limit group
```

Syntax: show rate-limit group

Output such as the following will display

```
rl-vlan-group 1
 vlan 10 to 15
```

This display shows the following information.

Table 10.2: Rate Limit VLAN Group Parameters

This Field...	Displays...
rl-vlan-group	The VLAN group whose contents are displayed.
vlan	VLANs contained in the VLAN group specified.

Displaying Rate Limit Policies per Interface

To display information about rate limit policies that are configured per interface, enter the following command:'

```
NI IMR640 Router# show rate-limit
```

Syntax: show rate-limit

Output such as the following will display


```
NI IMR640 Router(config-if-e10000-1/1)#show rate-limit

interface e 1/1
 rate-limit input 959904 2000000
 rate-limit output 2986368 2000000
```

This display shows the following information.

Table 10.3: Rate Limit Interface Parameters

This Field...	Displays...
rate-limit input	The average-rate and maximum burst rate configured for inbound traffic on the specified interface.
rate-limit output	The average-rate and maximum burst rate configured for outbound traffic on the specified interface.

Displaying Rate Limit Policies Configured in Policy Maps

To display information about rate limit policy maps, enter the following command:'

```
NI IMR640 Router# show policy-map
```

Syntax: show policy-map [map-name]

The <map-name> variable limits the display of policy map configuration information to the map specified. If this variable is not used, configuration information will be displayed for all policy maps configured on the router.

Output such as the following will display

```
NI IMR640 Router(config-policymap pmap1)#show policy-map

policy-map pmap1
  cir 106656       bps cbs 24000        bytes
  eir 53328        bps ebs 20000        bytes
  excess-priority 2 excess-dscp 43

policy-map pmap2
  cir 106656       bps cbs 24000        bytes
  eir 53328        bps ebs 30000        bytes
  excess-priority 1 excess-dscp 30
```

This display shows the following information.

Table 10.4: Rate Limit Policy Map Parameters

This Field...	Displays...
policy-map	The name of the policy map whose configuration is being displayed
cir	The value of the Committed Information Rate (CIR) configured for this policy map. Possible values are: 1 - 10000000000 bps.
cbs	The value of the Committed Burst Size (CBS) configured for this policy map. Possible values are: 1250 - 1250000000 bytes.
eir	The value of the Excess Information Rate (EIR) configured for this policy map. Possible values are: 1 - 10000000000 bps.

Table 10.4: Rate Limit Policy Map Parameters (Continued)

This Field...	Displays...
ebs	The value of the Excess Burst Size (EBS) configured for this policy map. Possible values are: 1250 - 1250000000 bytes.
excess-priority	The priority queue that traffic whose bandwidth requirements exceeds what is available in the CIR bucket and is sent to the EIR bucket is set to. Possible values are 0-3.
excess-dscp	The priority queue that traffic whose bandwidth requirements exceeds what is available in the CIR bucket and is sent to the EIR bucket is set to. Possible values are 0-63

Chapter 11
Configuring Rate Limiting on Other Foundry Devices

This chapter describes how to configure rate limiting on Foundry devices that do not have IronCore or JetCore modules.

NOTE: To configure rate limiting on an IronCore module, see “Configuring IronClad Rate Limiting (IronCore)” on page 8-1. To configure rate limiting on a JetCore Chassis device or the FastIron 4802, see “Configuring JetCore Rate Limiting (JetCore)” on page 9-1.

Fixed Rate Limiting on a FastIron Edge Switch (FES)

Software release 03.1.00 and later for the FastIron Edge Switch (FES) supports fixed rate limiting for inbound traffic, on individual ports. The fixed rate limiting is at line rate and occurs in hardware. Fixed rate limiting allows you to specify the maximum number of bits per second (bps) a port can receive. The port drops all traffic that exceeds the specified bps within a given one-second interval. Fixed rate limiting applies to all traffic on the rate limited port.

The rate you specify applies to each one-second interval. All traffic that exceeds the specified rate within a one-second interval is dropped. Unused bandwidth is not carried over from one interval to the next.

Configuring Rate Limiting

To configure rate limiting on a port, enter commands such as the following:

```
FES4802 Router(config)# interface ethernet 48
FES4802 Router(config-if-e100-48)# rate input fixed 10000000
```

These commands limit the average rate for inbound traffic on port 48 to 10,000,000 bps.

Syntax: [no] rate input fixed <average-rate> [payload-only]

The <average-rate> parameter specifies the number of bits per second (bps) the port can receive. The minimum rate that can be configured is 240,000 bps.

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. If this parameter is specified, then 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

Displaying the Fixed Rate Limiting Configuration

To display the fixed rate limiting configuration on the device, enter the following command:

```
FES4802 Switch(config-if-e100-21)#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 number of ports on which fixed rate limiting is configured, then provides the following information for each of the ports:

- The **Configured Input Rate** is the rate requested in the configuration.
- The **Actual Input Rate** is the rate provided by the hardware for the request.
- **Mode** will indicate if the payload-only option has been specified.
-

Rate Limiting on Terathon Devices

Terathon devices running Terathon software release 01.1.00 supports rate limiting. It provides line-rate rate limiting in hardware on inbound ports.

You can configure the Terathon device to use one of the following modes of rate limiting policies:

- Port-based for inbound and outbound ports– Limits the rate of inbound and outbound traffic on an individual physical port to a specified rate. Only one port-based inbound and outbound rate limiting policy can be applied to a port. (Refer to “Configuring Port-Based Rate Limiting For Inbound and Outbound Ports” on page 11-5.)
- Port-and-priority-based – Limits the rate on an individual hardware forwarding queue on an individual physical port. Only one port-and-priority-based rate limiting policy can be specified per priority queue for a port. This means that a maximum of four port-and-priority-based policies can be configured on a port. (Refer to

"Configuring a Port-and-Priority-Based Rate Limiting Policy" on page 11-6.)

- Port-and-VLAN-based – Limits the rate of packets tagged with a specific VLAN on an individual physical port. Only one rate can be specified for each VLAN. Up to 10 VLAN-based policies can be configured for a port. (Refer to "Configuring a Port-and-VLAN-Based Rate Limiting Policy" on page 11-6.)
- Port-and-ACL-based – Limits the rate of IP traffic on an individual physical port that matches the permit conditions in IP Access Control Lists (ACLs). You can use standard or extended IP ACLs. Standard IP ACLs match traffic based on source IP address information. Extended ACLs match traffic based on source and destination IP address and IP protocol information. Extended ACLs for TCP and UDP also match on source and destination TCP or UDP addresses. and protocol information. (Refer to "Configuring a Port-and-ACL-Based Rate Limiting Policy" on page 11-8.)

Software Release 02.1.00 adds the following enhancements to the rate limiting feature:

- Port-based for outbound ports – Limits the rate of outbound traffic on an individual physical port to a specified rate. Only one port-based outbound rate limiting policy can be applied to a port. (Refer to "Configuring Port-Based Rate Limiting For Inbound and Outbound Ports" on page 11-5.)
- Port-and-Layer 2 ACL-based – Limits the rate of traffic on an individual physical port that matches the permit conditions a Layer 2 ACL. (Refer to "Configuring Port-and-Layer 2 ACL-Based Rate Limiting" on page 11-9.)
- VLAN-and-priority based – Limits traffic on a physical port that is a member of a specified VLAN and has been assigned to specified forwarding queues. (Refer to "Configuring VLAN-and-Priority-Based Rate Limiting" on page 11-6.)
- VLAN group based – Limits the traffic for a group of VLANs. Members of a VLAN group share the specified bandwidth defined in the rate limiting policy that has been applied to that group. (Refer to "Configuring VLAN Group Based Rate Limiting" on page 11-7.)
- Port-and-IPV6 ACL-based – Limits the rate of traffic on an individual physical port that matches the permit conditions of IPV6 ACL. These policies can be applied to inbound traffic only. (Refer to "Configuring Port-and-IPv6 ACL-Based Rate Limiting" on page 11-10.)
- Filtering traffic denied by a rate limiting ACL – Drops traffic that matched an ACL deny filter in a port-and-ACL based rate limiting policy. (Refer to "Filtering Traffic Denied by a Rate Limiting ACL" on page 11-10.)
- New command to display rate limiting policies – Displays rate limiting policies that have been configured for a device, an interface, or a VLAN group. (Refer to "Displaying Rate Limiting Policies" on page 11-10 and "Displaying Rate Limit VLAN Groups" on page 11-12.)

This section presents all the rate limiting policies available on Terathon devices. Except for port-based rate limiting policies, all rate limiting policy types can be applied only to inbound ports.

Rate Limiting Parameters and Algorithm

A rate limiting policy specifies two parameters: average rate and maximum burst. These parameters are used to configure credits and credit totals.

Average Rate

The *Average Rate* is the maximum number of bits a port is allowed to receive during a one-second interval. The rate of the traffic that matches the rate limiting policy will not exceed the average rate.

The Average Rate represents a percentage of an interface's line rate (bandwidth), expressed in bits per second (bps). It cannot be smaller than 515,624 bits per second (bps) and it cannot be larger than the port's line rate.

Average Rate must be entered in multiples of 515,624 bps. If you enter a number that is not a multiple of 515,624, the software adjusts the rate down to the lowest multiple of the number so that the calculation of credits does not result in a remainder of a partial Credit. For example, if you enter 600,000 bps, the value will be adjusted to 515,624 bps. The adjusted rate is sometimes called the *adjusted average rate*.

Maximum Burst

Maximum burst provides a higher than average rate to traffic that meet the rate limiting criteria. When the traffic on the port is less than the specified average rate, the rate limiting policy can accumulate credits up to a maximum, as

specified in the maximum burst value. The accumulated credit allows traffic to pass through the port for a short period of time, at a rate higher than the average rate. The time period is determined by the amount of credit accumulated and the rate of traffic passing through the port.

The maximum burst rate cannot be smaller than 65536 bits.

Credits and Credit Total

Each rate limiting policy is assigned a class. A *class* uses the average rate and maximum allowed burst in the rate limiting policy to calculate credits and credit totals.

Credit size is measured in bytes. A credit is a forwarding allowance for a rate-limited port, and is the smallest number of bytes that can be allowed during a rate limiting interval. The minimum credit size can be 1 byte.

During a rate limiting interval, a port can send or receive only as many bytes as the port has Credits for. For example, if an inbound rate limiting policy results in a port receiving two credits per rate limiting interval, the port can send or receive a maximum of 2 bytes of data during that interval.

The credit size is calculated using the following algorithm:

Credit = (Average rate in bits per second)/(8*64453)

One second is divided into 64,453 intervals. In each interval, the number of bytes equal to the credit size is added to the running total of the class. The running total of a class represents the number of bytes that can be allowed to pass through without being subject to rate limiting.

The second parameter is the maximum *credit total*, which is also measured in bytes. The maximum credit total is calculated using the following algorithm.

Maximum credit total = (Maximum burst in bits)/8

The running total can never exceed the maximum credit total. When packets arrive at the port, a class is assigned to the packet, based on the rate limiting policies. If the running total of the class is less than the size of the packet, then the packet is dropped. Otherwise, the size of the packet is subtracted from the running total and the packet is forwarded. If there is no traffic that matches rate limiting criteria, then the running total can increase until it reaches the maximum credit total.

Configuration Considerations

- Except for port-based rate limiting policies, all rate limiting policy types can be applied only to inbound ports on Terathon devices.
- Only one type of inbound rate limiting policy can be applied on a physical port. For example, you cannot apply inbound port-and-ACL-based and inbound port-based rate limiting policies on the same port.
- Outbound port-based rate limiting policy can be combined with any type of inbound rate limiting policy.
- When a port-and-VLAN-based rate limiting policy is applied to a port, all the ports controlled by the same packet processor are rate limited for that VLAN. You cannot apply a port-and-VLAN-based rate limiting policy on another port of the same packet processor for the same VLAN ID.
- Any VLAN-based rate limiting can limit only tagged packets that match the VLAN ID specified in the policy. Untagged packets are not subject to rate limiting.
- The average rate in a rate limiting policy cannot be less than 515,624 bits per second, must be in multiples of 515,624, and cannot be more than the port's line rate.
- The maximum burst in a rate limit policy can be less than the average rate, but cannot be less than 65536 bits and cannot be more than the port's line rate.
- Control packets are not subject to rate limiting.
- You cannot apply Layer4 ACL-based rate limiting policy on a physical port that is a member of a virtual routing interface.
- You cannot create a trunk if any of the physical ports that are members of the trunk has a rate limiting policy.
- You cannot apply a Layer 2 ACL-based rate limit policy and a Layer 4 ACL-based rate limit policy on a port at

the same time.

- A Layer 4 ACL-based rate limiting policy applies only to Layer 3 traffic.
- The total number of source MAC-and-VLAN based, any ACL-based, and any VLAN-based rate limiting policies on ports controlled by the same packet processor cannot exceed:
  - 126 on a 4 x 10G Interface module
  - 117 on a 40 x 1G Interface module
  - 107 on a 60 x 1G Interface module
- For any type of priority based rate limiting policy on a port: If the rates of the policies are the same, then the priorities are combined into one group. For example:

  ```
  BigIron MG8 (config-if-1/1)#rate-limit in priority q1 500000000 750000000
  BigIron MG8 (config-if-1/1)#rate-limit in priority q2 500000000 750000000
  ```

  These two policies will be combined and displayed as one policy:

  ```
  BigIron MG8 (config-if-1/1)#rate-limit in priority q1 q2 500000000 750000000
  ```

  All the traffic for hardware forwarding queues q1 and q2 will be rate limited individually to an average rate of 500Mbps with a maximum burst size of 750Mbits, even if the queues are combined into one policy.
- Certain features such as FDP, CDP, UDLD and LACP that make the port run in dual mode can cause traffic to be rate limited to less than the expected average rate. When the port is in dual mode, all incoming or outgoing packets are treated as tagged. An extra 4 bytes is added to the length of the packet to account for the tag, thus causing the average rate to be less than the expected average rate. Ports in dual mode are assumed to be tagged ports for rate limiting purpose.

Configuring Rate Limiting on Terathon Devices

The following sections show examples of how to configure each rate limiting policy type.

Configuring Port-Based Rate Limiting For Inbound and Outbound Ports

BigIron MG8 and NetIron 40G software release 01.1.00 introduced rate limiting features for inbound ports. Software release 02.1.00 adds port-based rate limiting to outbound ports.

Port-based rate limiting limits the rate on an individual physical port to a specified rate.

To configure port-based rate limiting policy for outbound ports, enter commands such as the following at the interface level:

```
BigIron MG8(config)# interface ethernet 1/1
BigIron MG8(config-if-1/1)# rate-limit out 500000000 750000000
Average rate is adjusted to 499639656 bits per second
```

The commands configure a rate limiting policy for outbound traffic on port 1/1. The policy limits the average rate of all outbound traffic to 500 Mbps with a maximum burst size of 750 Mbps.

The complete syntax for configuring a port-based rate limiting policy is:

Syntax: [no] rate-limit in | out <average-rate> <maximum-burst>

The **in** parameter applies the policy to traffic on inbound ports, while **out** applies to outbound ports.

Only one inbound and one outbound port-based rate limiting policy can be applied to a port.

The <average-rate> parameter specifies the maximum rate allowed on a port during a one-second interval. The software automatically adjusts the number you enter to the lower multiple of 515,624 bps. Refer to the section “Average Rate” on page 11-3 for more details.

The <maximum-burst> parameter specifies the extra bits above the average-rate that traffic can have. Refer to the section “Maximum Burst” on page 11-3 for more details.

Configuring a Port-and-Priority-Based Rate Limiting Policy

To configure port-and-priority based rate limiting policy:

```
BigIron MG8(config)# interface ethernet 1/1
BigIron MG8(config-if-1/1)# rate-limit in priority q0 q2 500000000 750000000
Average rate is adjusted to 499639656 bits per second
```

These commands configure a rate limiting policy for inbound port 1/1 that limits the average rate of all inbound traffic for hardware forwarding queues q0 and q2. Traffic on each hardware forwarding queue is limited to an average rate of 500 Mbps with a maximum burst size of 750 Mbits.

Syntax: [no] rate-limit in priority q0 | q1 | q2 | q3 <average-rate> <maximum-burst>

The **in** parameter applies the policy to traffic on inbound ports.

The **priority q0 | q1 | q2 | q3** parameter specifies the hardware forwarding queue to which the policy applies. The device prioritizes the queues from **q0** (normal priority) to **q3** (highest priority). Only one rate can be specified per priority queue for a port.

The <average-rate> parameter specifies the maximum rate allowed on a port during a one-second interval. The software automatically adjusts the number you enter to the lower multiple of 515,624 bps. Refer to the section "Average Rate" on page 11-3 for more details.

The <maximum-burst> parameter specifies the extra bits above the average-rate that traffic can have. Refer to the section "Maximum Burst" on page 11-3 for more details.

Configuring a Port-and-VLAN-Based Rate Limiting Policy

To configure a port-and-VLAN based rate limiting policy, enter commands such as the following:

```
BigIron MG8(config)# interface ethernet 1/1
BigIron MG8(config-if-1/1)# rate-limit in vlan 10 500000000 750000000
Average rate is adjusted to 499639656 bits per second
BigIron MG8(config-if-1/1)# rate-limit in vlan 20 100000000 200000000
Average rate is adjusted to 99515432 bits per second
```

These commands configure two rate limiting policies that limit the average rate of all inbound traffic on port 1/1 with VLAN tag 10 and 20. The first policy limits packets with VLAN tag 10 to an average rate of 500 Mbps with a maximum burst size of 750 bits. The second policy limits packets with VLAN tag 20 to an average rate of 100 Mbps with a maximum burst size of 200 Mbits. Tagged packets belonging to VLANs other than 10 and 20 and untagged packets are not subject to rate limiting.

Syntax: [no] rate-limit in vlan <vlan-number> <average-rate> <maximum-burst>

The **in** parameter applies the policy to traffic on inbound ports.

The vlan <vlan-number> parameter species the VLAN ID to which the policy applies. Refer to "Configuration Considerations" on page 11-4 to determine the number of rate limiting policies that can be configured on a device.

The <average-rate> parameter specifies the maximum rate allowed on a port during a one-second interval. The software automatically adjusts the number you enter to the lower multiple of 515,624 bps. Refer to the section "Average Rate" on page 11-3 for more details.

The <maximum-burst> parameter specifies the extra bits above the average-rate that traffic can have. Refer to the section "Maximum Burst" on page 11-3 for more details.

Configuring VLAN-and-Priority-Based Rate Limiting

VLAN-and-priority based rate limiting limits traffic on a physical port that is a member of a specified VLAN and has been assigned to specified forwarding queues. For example, you can configure a rate limiting policy for inbound traffic on port 1/1. The policy limits the average rate of all inbound packets with VLAN tag 10 destined for hardware forwarding queues q0 and q2 to an average rate of 500Mbps for each queue with a maximum burst size of 750 Mbits for each queue. Enter commands such as the following:

```
BigIron MG8(config)# interface ethernet 1/1
BigIron MG8(config-if-1/1)# rate-limit in vlan 10 pri q0 q2 500000000 750000000
```


Syntax: [no] rate-limit in vlan <number> priority <queue> <average-rate> <maximum-burst>

The **in** parameter applies the policy to traffic on inbound ports.

Enter the VLAN ID for the **vlan** <number> parameter.

The **priority q0 | q1 | q2 | q3** parameter specifies the hardware forwarding queue to which the policy applies. The device prioritizes the queues from **q0** (normal priority) to **q3** (highest priority).

The <average-rate> parameter specifies the maximum rate allowed on a port during a one-second interval. The software automatically adjusts the number you enter to the lower multiple of 515,624 bps. Refer to the section "Average Rate" on page 11-3 for more details.

The <maximum-burst> parameter specifies the extra bits above the average-rate that traffic can have. Refer to the section "Maximum Burst" on page 11-3 for more details.

Configuring VLAN Group Based Rate Limiting

A rate limiting policy can be applied to a VLAN group. VLANs that are members of a VLAN group share the specified bandwidth defined in the rate limiting policy applied to that group.

To configure a rate limiting policy for a VLAN group, do the following:

1. Define the VLANs that you want to place in a rate limiting VLAN group.
2. Define a rate limiting VLAN group. This VLAN group is specific to the rate limiting feature. Enter commands such as the following:

   ```
   BigIron MG8(config)# rl-vlan-group 10
   BigIron MG8(config-vlan-rate-group)# vlan 3 5 to 7 10
   BigIron MG8(config-vlan-rate-group)# exit
   ```

   The commands assign VLANs 3, 5,6, 7, and 10 to rate limiting VLAN group 10.

   Syntax: [no] rl-vlan-group <vlan-group-number>

   Syntax: [no] vlan <vlan-number> [to <vlan-number]

   The **rl-vlan-group** command takes you to the VLAN group rate limiting level. Enter the ID of the VLAN group that you want to create or update by entering a value for <vlan-group-number>.

   Use the vlan command to assign or remove VLANs to the rate limiting VLAN group. You can enter the individual VLAN IDs or a range of VLAN IDs.

3. Create a policy for the VLAN group and apply it to the interface you want. Enter commands such as the following:

   ```
   BigIron MG8(config)# int e 1/1
   BigIron MG8(config-if-1/1)# rate-limit in group 10 500000000 750000000
   ```

   The command applies the rate limiting policy for rate limiting VLAN group 10 on port 1/1. This policy limits all traffic tagged with VLANs 3, 5, 6, 7, or 10 to an average rate of 500 Mbps with a maximum burst size of 750 Mbits.

   Syntax: rate-limit in group <group-number> <average-rate> <maximum-burst>

   The **in** parameter indicates that the policy is for incoming traffic.

   Enter the rate limiting VLAN group ID for the **group** <group-number> parameter.

   The <average-rate> parameter specifies the maximum rate allowed on a port during a one-second interval. The software automatically adjusts the number you enter to the lower multiple of 515,624 bps. Refer to the section "Average Rate" on page 11-3 for more details.

   The <maximum-burst> parameter specifies the extra bits above the average-rate that traffic can have. Refer to the section "Maximum Burst" on page 11-3 for more details.

4. If you want to apply a rate limiting policy to a VLAN group whose traffic are prioritized by hardware forwarding queues, enter commands such as the following:


```
BigIron MG8(config)# int e 1/1
BigIron MG8(config-if-1/1)# rate limit in group 10 priority q1 q2 500000000
750000000
```

The command applies the rate limiting policy for rate limiting VLAN group 10 on port 1/1. This policy limits all traffic tagged with VLANs 3, 5, 6, 7, or 10 on each hardware forwarding queue. Rate for q1 is rate limited to an average rate of 500 Mbps with a maximum burst size of 750 Mbits. Rate for q2 is also rate limited to an average rate of 500 Mbps with a maximum burst size of 750 Mbits.

Configuration Considerations

When configuring VLAN group based rate limiting policies, consider the following rules:

- A rate limit VLAN group must have at least one VLAN member before it can be used in a rate limit policy. The list cannot be empty if it is being used in a rate limiting policy.
- A rate limit VLAN group cannot be deleted if it is being used in a rate limiting policy.
- If a rate limit policy for a VLAN group is applied to a port, the group cannot be used in any other rate limiting policies applied to other ports that are controlled by the same packet processor.
- A VLAN can be member of multiple rate limit VLAN groups, but two groups with common members cannot be applied on ports controlled by the same packet processor.
- VLAN-based rate limiting and VLAN groups based rate limiting policies can be applied on the same ports or ports controlled by the same packet processor as long as there are no common VLANs in the policies.

Configuring a Port-and-ACL-Based Rate Limiting Policy

You can use standard or extended IP ACLs for port-and-ACL-based rate limiting.

- Standard IP ACLs match traffic based on source IP address information.
- Extended ACLs match traffic based on source and destination IP addresses and IP protocol information. Extended ACLs for TCP and UDP protocol must also match on source and destination IP addresses and TCP or UDP protocol information.
- You can apply an ACL ID to a port-and-ACL-based rate limiting policy even before you define the ACL. The rate limiting policy does not take effect until the ACL is defined.
- It is not necessary to remove an ACL from a port-and-ACL-based rate limiting policy before deleting the ACL.

NOTE: Port-and-ACL-based rate limiting is supported for traffic on inbound ports only.

To configure port-and-ACL-based rate limiting policies, enter commands such as the following:

```
BigIron MG8(config)#access-list 50 permit host 1.1.1.2
BigIron MG8(config)#access-list 50 deny host 1.1.1.3
BigIron MG8(config)#access-list 60 permit host 2.2.2.3
BigIron MG8(config-if-1/1)# rate-limit in access-group 50 500000000 750000000
Average rate is adjusted to 499639656 bits per second
BigIron MG8(config-if-1/1)# rate-limit in access-group 60 100000000 200000000
Average rate is adjusted to 99515432 bits per second
```

These commands first configure access-list groups that contain the ACLs that will be used in the rate limiting policy. Use the **permit** condition for traffic that will be rate limited. Traffic that match the **deny** condition are not subject to rate limiting and allowed to pass through. Refer to “Filtering Traffic Denied by a Rate Limiting ACL” on page 11-10 for information on how to drop traffic that matches deny conditions. .

Next, the commands configure two rate limiting policies on port 1/1. The policies limit the average rate of all inbound IP traffic that match the permit rules of ACLs 50 and 60. The first policy limits the rate of all permitted IP traffic from host 1.1.1.2 to an average rate of 500 Mbps with a maximum burst size of 750 bits. Rate of all traffic from host 1.1.1.3 is not subject to rate limiting since it is denied by ACL 50; it is merely forwarded on the port.

The second policy limits the rate of all IP traffic from host 2.2.2.3 to an average rate of 100 Mbps with a maximum burst size of 200 Mbits.

All IP traffic that does not match ACLs 50 and 60 are not subject to rate limiting.

Syntax:

The **in** parameter applies the policy to traffic on inbound ports.

The **access-group**, group-number> parameter specifies the group number to which the ACLs used in the policy belong.

NOTE: An ACL must exist in the configuration before it can take effect in a rate limiting policy.

Refer to the "Configuration Considerations" on page 11-4 regarding the number of ACL-based rate limiting policies that can be configured.

The <average-rate> parameter specifies the maximum rate allowed on a port during a one-second interval. The software automatically adjusts the number you enter to the lower multiple of 515,624 bps. Refer to the section "Average Rate" on page 11-3 for more details.

The <maximum-burst> parameter specifies the extra bits above the average-rate that traffic can have. Refer to the section "Maximum Burst" on page 11-3 for more details.

Configuring Port-and-Layer 2 ACL-Based Rate Limiting

The port-and-Layer 2 ACL-based rate limiting limits the rate of traffic on individual physical ports that match the permit conditions a Layer 2 ACL. For example,

```
BigIron MG8(config)# access-list 400 deny any any any etype arp
BigIron MG8(config)# access-list 400 deny any any any etype ipv4
BigIron MG8(config)# access-list 400 permit any any 100

BigIron MG8(config)# interface ethernet 1/1
BigIron MG8(config-if-1/1)# rate-limit in access-group 400 100000000 200000000
Average rate is adjusted to 99515432 bits per second
```

These commands first configure access-list group 400. This group contains the ACLs that will be used in the rate limiting policy. Use the **permit** condition for traffic that will be rate limited. Traffic that match the **deny** condition are not subject to rate limiting.

The next set of commands configures a rate limiting policies on port 1/1. The policies limit the average rate of all inbound IP traffic that match the permit rules of ACL 400 to an average rate of 100 Mbps with a maximum burst size of 200 Mbits. Traffic denied by ACL 400 is merely forwarded on the port.

Syntax: [no] rate-limit in access-group <number> <average-rate> <maximum-burst>

The **in** parameter applies the policy to traffic on inbound ports.

The **access-group** <number> parameter identifies the Layer 2 ACL used to permit or deny traffic on a port. Permitted traffic is subject to rate limiting.

NOTE: Port-and Layer 2 ACL-based rate limiting and Port-and-Layer 4 ACL-based rate limiting cannot be applied on a port at the same time.

The <average-rate> parameter specifies the maximum rate allowed on a port during a one-second interval. The software automatically adjusts the number you enter to the lower multiple of 515,624 bps. Refer to the section "Average Rate" on page 11-3 for more details.

The <maximum-burst> parameter specifies the extra bits above the average-rate that traffic can have. Refer to the section "Maximum Burst" on page 11-3 for more details.

Configuring Port-and-IPv6 ACL-Based Rate Limiting

This release supports port-and-IPv6 ACL-based rate limiting. The port-and-IPV6 ACL-based rate limiting limits the rate of traffic on individual physical ports that match the permit conditions of an IPV6 ACL. Traffic that matches the deny condition is not subject to rate limiting.

For example, the following commands in the Global Config mode configure the IPv6 access-list: "ipv6-acl" to permit any traffic from the 10:10::0:0/64 network and deny all other traffic.

```
BigIron(config)# ipv6 access-list ipv6-acl
BigIron(config-ipv6-access-list ipv6-acl)# permit ipv6 10:10::0:0/64 any
BigIron(config-ipv6-access-list ipv6-acl)# deny ipv6 any any
```

The following configuration creates a rate limiting policy on port 1/1. The policy limits the average rate of all inbound IP traffic that matches the permit rules of ACL “ipv6-rl” to an average rate of 100 Mbps with a maximum burst size of 200 Mbits. Traffic denied by ACL “ipv6-rl” is forwarded on the port.

```
BigIron(config)# interface ethernet 1/1
BigIron(config-if-1/1)# rate-limit in ipv6-named-access-group ipv6-rl 100000000
200000000

Average rate is adjusted to 99515432 bits per second
```

Syntax: [no] rate-limit in ipv6-named-access-group <name> <average-rate> <maximum-burst>

The **in** parameter applies the policy to traffic on inbound ports.

The **ipv6-named-access-group** <name> parameter identifies the IPv6 ACL used to permit or deny traffic on a port. Permitted traffic is subject to rate limiting. Denied traffic is forwarded on the port.

The <average-rate> parameter specifies the maximum rate allowed on a port during a one-second interval.

The <maximum-burst> parameter specifies the extra Mbits above the average-rate that traffic can have.

Filtering Traffic Denied by a Rate Limiting ACL

When you use a Layer 2 ACL-based or Layer 4 ACL-based rate limiting policy, traffic permitted by the ACL is subject to rate limiting; however, traffic denied by the ACL is simply forwarded on the port. With the strict ACL feature, you can configure a port to drop traffic that is denied by the rate limiting ACL instead of forwarding it.

NOTE: Once you configure a Layer 2 ACL-based or Layer 4 ACL-based rate limiting policy on a port, you cannot configure a regular (traffic filtering) ACL on the same port. To filter this type of traffic, you must enable the strict ACL feature.

To enable the device to drop traffic that is denied by a rate limiting ACL, enter the following command at the configuration level for the port:

```
BigIron MG8(config-if-1/1)# rate-limit strict-acl
```

Syntax: [no] rate-limit strict-acl

Displaying Rate Limiting Policies

The **show rate-limit** command has been added to display the rate limiting policies that have been configured on an interface.

For example, to display rate limiting policy on a device, enter the following command:

```
BigIron MG8(config)# show rate-limit
interface e 1/1
 rate-limit input group 3 8765608 9000000
 rate-limit input group 10 priority q1 515624 1000000
 rate-limit input group 10 priority q0 q2 2578120 3000000
interface e 1/2
 rate-limit input 8765608 9000000
interface e 1/3
 rate-limit input vlan-id 5 515624 1000000
```

To display rate limiting policy on a device with counters, enter the following command:

```
BigIron MG8(config)# show rate-limit counters
interface e 1/1
 rate-limit input group 3 8765608 9000000
  Pkts fwd: 20 Pkts drop: 10 Total: 30
 rate-limit input group 10 priority q1 515624 1000000
  Pkts fwd: 90 Pkts drop: 15 Total: 105
 rate-limit input group 10 priority q0 q2 2578120 3000000
  Pkts fwd: 221 Pkts drop: 11 Total: 232
 rate-limit input group 20 priority q1 q2 q3 515624 1000000
  Pkts fwd: 0 Pkts drop: 0 Total: 0
interface e 1/2
 rate-limit input 8765608 9000000
  Pkts fwd: 440 Pkts drop: 20 Total: 460
interface e 1/3
 rate-limit input vlan-id 5 515624 1000000
  Pkts fwd: 0 Pkts drop: 0 Total: 0
```

To display rate limiting policy on a BigIron MG8 or NetIron 40G running software release 02.2.00 and later:

```
BigIron MG8(config)# show rate-limit counters

interface e 1/2 rate-limit
 input 8765608 9000000
 Bytes fwd: 440 Bytes drop: 20 Total: 460
```

The byte accounting statistics are displayed above in **bold**. The byte count includes the preamble and the minimum inter-frame gap in Ethernet.

To display the rate limiting policies on interface 1/3, enter the following command:

```
BigIron MG8(config)# show rate-limit interface 1/3
interface e 1/3
 rate-limit input vlan-id 5 515624 1000000
```


You can also display rate limiting policies for an interface that includes counters by entering the following command:

```
BigIron MG8(config)# show rate-limit counters interface 1/4
interface e 1/4
 rate-limit input priority q1 8765608 9000000
  Pkts fwd: 200 Pkts drop: 150 Total: 350
```

Syntax: show rate-limit [counters] [interface <slot-number/port-number>]

For inbound rate limiting policies, specify the **counters** parameter if you want counters to be included in the display. Counters show the estimated number of packets that matched a rate limiting policy and were either forwarded or dropped, based on the availability of credit. If you do not use this parameter, the counters are not included in the display.

Outbound port rate limiting policies have no counters.

Use the **interface** <slot-number/port-number> to display rate limiting policies for a specific interface.

Displaying Rate Limit VLAN Groups

To display the rate limit VLAN groups and their members, enter the following command:

```
BigIron MG8#show rate-limit group
rl-vlan-group 3
 vlan 2 to 3
rl-vlan-group 10
 vlan 25 29 to 40 42 100 to 2000
```

To display VLAN members of a specific rate limit VLAN group, enter a command such as the following:

```
BigIron MG8#show rate-limit group 3
rl-vlan-group 3
 vlan 2 to 3
```

Syntax: show rate-limit group <group-number>

Specify the rate limit group number for the **group** <group-number> parameter.

Changes to Rate Limiting Counters in Terathon IronWare Release 02.2.00

In Terathon IronWare software release 02.2.00 and later for the BigIron MG8 and NetIron 40G, rate limiting counters have been changed to count bytes instead of packets. The CLI remains the same. The display will show bytes forwarded and dropped instead of packets forwarded and dropped.

```
BigIron MG8(config)# show rate-limit counters
```

```
interface e 1/2 rate-limit
 input 8765608 9000000
 Bytes fwd: 440 Bytes drop: 20 Total: 460
```

The byte accounting statistics are displayed above in **bold**. The byte count includes the preamble and the minimum inter-frame gap in Ethernet.

Chapter 12
Configuring IP

This chapter describes the Internet Protocol (IP) parameters on Foundry Layer 2 Switches and Layer 3 Switches and how to configure them. After you add IP addresses and configure other IP parameters, see the following chapters for configuration information for the IP routing protocols:

- “Configuring RIP” on page 13-1
- “Configuring OSPF” on page 15-1
- “Configuring BGP4” on page 16-1

To configure and monitor IP, see the following sections:

- “Basic IP Parameters and Defaults – Layer 3 Switches” on page 12-9
- “Basic IP Parameters and Defaults – Layer 2 Switches” on page 12-17
- “Configuring IP Parameters – Layer 3 Switches” on page 12-19
- “Configuring IP Parameters – Layer 2 Switches” on page 12-97
- “Displaying IP Configuration Information and Statistics” on page 12-104

NOTE: The NetIron 400 and NetIron 800 are chassis-based Internet backbone routers. References to chassis-based Layer 3 Switches also apply to the NetIron 400 and NetIron 800 unless otherwise noted.

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 12-19 to add IP addresses, then see one or more of the following to enable and configure the route exchange protocols:
  - “Configuring RIP” on page 13-1
  - “Configuring OSPF” on page 15-1
  - “Configuring BGP4” on page 16-1
- If you are configuring a Layer 2 Switch, see “Configuring the Management IP Address and Specifying the Default Gateway” on page 12-97 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)
  - Foundry Standby Router Protocol (FSRP)

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 12-104.

Layer 3 Switches

Foundry Layer 3 Switches allow you to configure IP addresses on the following types of interfaces:

- Ethernet ports
- Packet over SONET (POS) 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 “Displaying and Modifying System Parameter Default Settings” section in the “Configuring Basic Features” chapter of the *Foundry Switch and Router Installation and Basic Configuration Guide* or the *Foundry BigIron Mg8 Switch Installation and Basic Configuration Guide*.

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 12.1 shows how an IP packet moves through a Foundry Layer 3 Switch.

Figure 12.1 IP Packet flow through a Foundry Layer 3 Switch

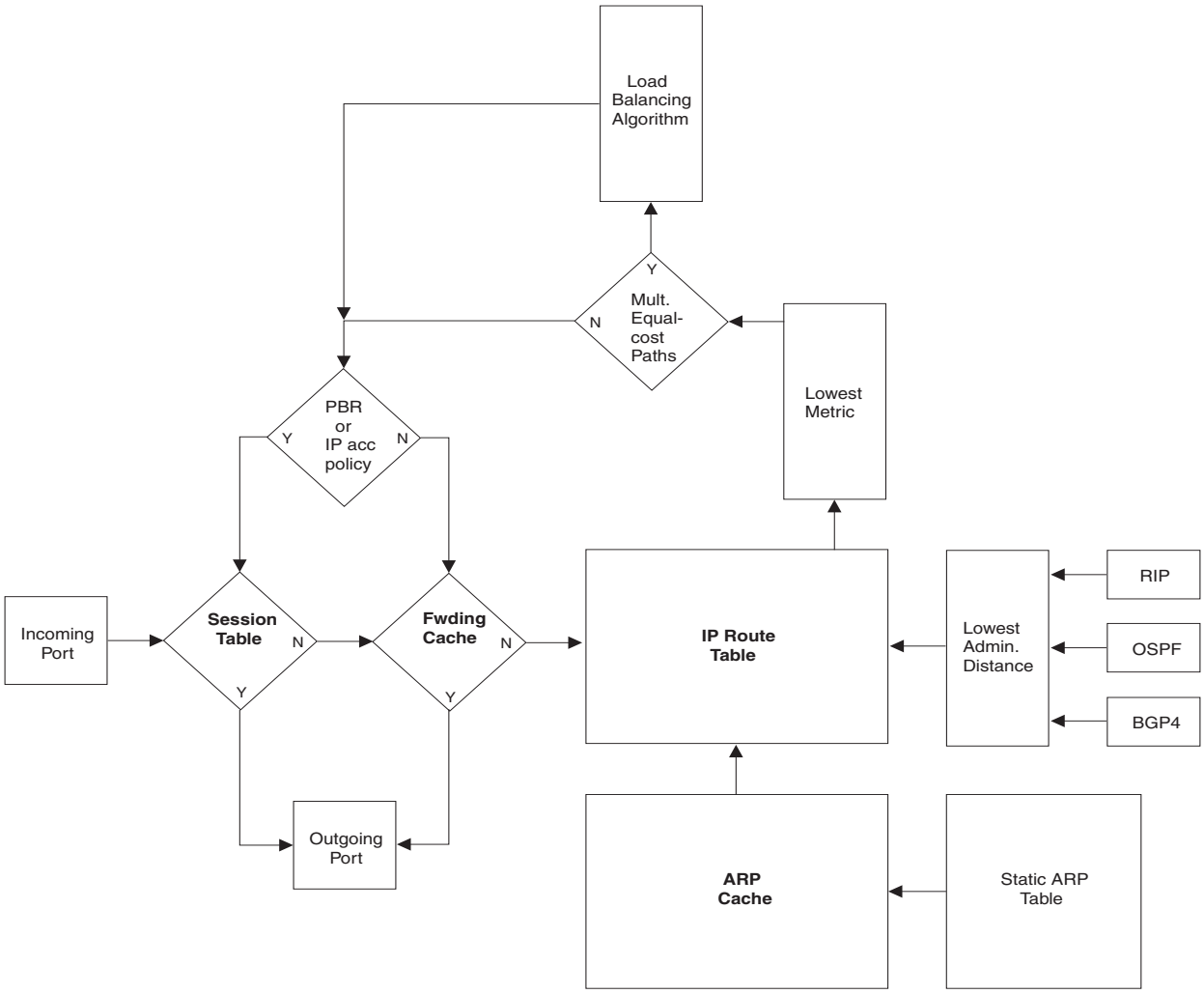


Figure 12.1 shows the following packet flow:

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).
    - If the running-config contains a Policy-Based Routing (PBR) definition or 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 a PBR definition or 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.

1.The filter can be an Access Control List (ACL) or an IP access policy.

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 12-113 – Layer 3 Switch
- "Displaying the Static ARP Table" on page 12-115 – Layer 3 Switch only
- "Displaying ARP Entries" on page 12-130 – Layer 2 Switch

To configure other ARP parameters, see the following:

- "Configuring ARP Parameters" on page 12-43 – Layer 3 Switch only

To increase the size of the ARP cache and static ARP table, see the following:

- For dynamic entries, see the "Displaying and Modifying System Parameter Default Settings" section in the "Configuring Basic Features" chapter of the *Foundry Switch and Router Installation and Basic Configuration Guide* or the *Foundry BigIron Mg8 Switch Installation and Basic Configuration Guide.* 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 12-48 – 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 12-118 – Layer 3 Switch only

To configure a static IP route, see the following:

- "Configuring Static Routes" on page 12-54 – Layer 3 Switch only

To clear a route from the IP route table, see the following:

- "Clearing IP Routes" on page 12-121 – Layer 3 Switch only

To increase the size of the IP route table for learned and static routes, see the "Displaying and Modifying System Parameter Default Settings" section of the "Configuring Basic Features" chapter of the *Foundry Switch and Router Installation and Basic Configuration Guide* or the *Foundry BigIron Mg8 Switch Installation and Basic Configuration Guide*.

- 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.

NOTE: The forwarding cache on Foundry Layer 2 Switches is used only for IP router acceleration (Layer 3 switching). See the "Enabling IP or IPX Router Acceleration" section of the "Configuring Basic Features" chapter of the *Foundry Switch and Router Installation and Basic Configuration Guide*.

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 12-116.

NOTE: You cannot add static entries to the IP forwarding cache, although Chassis Layer 3 Switches do have options to optimize the cache and increase the number of entries the cache can contain. See "Optimizing the IP Forwarding Cache" on page 12-79 and the "Displaying and Modifying System Parameter Default Settings" section of the "Configuring Basic Features" chapter of the *Foundry Switch and Router Installation and Basic Configuration Guide* or the *Foundry BigIron Mg8 Switch Installation and Basic Configuration Guide.*

To increase the size of the IP forwarding cache, see the "Displaying and Modifying System Parameter Default Settings" section of the "Configuring Basic Features" chapter of the *Foundry Switch and Router Installation and Basic Configuration Guide* or the *Foundry BigIron Mg8 Switch Installation and Basic Configuration Guide*. The ip-cache parameter controls the size of the IP forwarding cache.

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:

- Policy-Based Routing (PBR)
- Layer 4 Quality-of-Service (QoS) policies
- IP access policies

To increase the size of the session table, see the "Displaying and Modifying System Parameter Default Settings" section of the "Configuring Basic Features" chapter of the *Foundry Switch and Router Installation and Basic Configuration Guide* or the *Foundry BigIron Mg8 Switch Installation and Basic Configuration Guide*. 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 13-1
- "Configuring OSPF" on page 15-1
- "Configuring BGP4" on page 16-1

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 14-1.

NOTE: Foundry Layer 2 Switches support IGMP and can forward IP multicast packets. See the “Configuring IP Multicast Traffic Reduction” chapter in the *Foundry Switch and Router Installation and Basic Configuration Guide*.

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.
- Foundry Standby Router Protocol (FSRP) – A Foundry router redundancy protocol developed before VRRP and VRRPE that provides some of the features of VRRP and some of the features of VRRPE. You can use FSRP 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 19-1.
- Virtual Router Redundancy Protocol (VRRP) – see “Configuring VRRP and VRRPE” on page 19-1.
- Foundry Standby Router Protocol (FSRP) – see “Configuring FSRP” on page 21-1

Network Address Translation

Foundry’s Chassis Layer 3 Switches support Network Address Translation (NAT). NAT enables private IP networks that use nonregistered IP addresses to connect to the Internet. Configure NAT on a Foundry Layer 3 Switch that is placed at the border of an inside network and an outside network (such as the Internet). NAT translates the internal local addresses to globally unique IP addresses before sending packets to the outside network.

For configuration information, see “Network Address Translation” on page 18-1.

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. ACLs also provide the input for Policy-Based Routing (PBR), which allows you to selectively modify and route IP packets based on their source IP address.

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, see the following:

- "Access Control List" on page 6-1
- "Policies and Filters" on page B-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 13-1
  - Open Shortest Path First (OSPF) – see "Configuring OSPF" on page 15-1
  - Border Gateway Protocol version 4 (BGP4) – see "Configuring BGP4" on page 16-1
- Multicast protocols
  - Internet Group Membership Protocol (IGMP) – see "Changing Global IP Multicast Parameters" on page 14-2
  - Protocol Independent Multicast Dense (PIM-DM) – see "PIM Dense" on page 14-13
  - Protocol Independent Multicast Sparse (PIM-SM) – see "PIM Sparse" on page 14-24
  - Distance Vector Multicast Routing Protocol (DVMRP) – see "DVMRP Overview" on page 14-65
- Router redundancy protocols
  - Virtual Router Redundancy Protocol Extended (VRRPE) – see "Configuring VRRP and VRRPE" on page 19-1.
  - Virtual Router Redundancy Protocol (VRRP) – see "Configuring VRRP and VRRPE" on page 19-1.
  - Foundry Standby Router Protocol (FSRP) – see "Configuring FSRP" on page 21-1

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 12.1 lists the IP global parameters for Layer 3 Switches.

Table 12.1: 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.	12-104
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.	12-40
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	12-33
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	12-43

Table 12.1: 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	12-44
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 12.2 on page 12-14.	Ten minutes	12-45
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	12-45
Static ARP entries	An ARP entry you place in the static ARP table. Static entries do not age out.	No entries	12-46
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	12-50
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 12.2 on page 12-14.	Disabled	12-50
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.	12-52
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	12-51

Table 12.1: IP Global Parameters – Layer 3 Switches (Continued)

Parameter	Description	Default	See page...
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 • Redirect messages **Note**: You also can enable or disable ICMP Redirect messages on an individual interface basis. See Table 12.2 on page 12-14.	Enabled	12-52 12-54
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 12.2 on page 12-14.	Disabled	12-87
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	12-89
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	12-90
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	12-96
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	12-23
DNS default gateway addresses	A list of gateways attached to the router through which clients attached to the router can reach DNSs.	None configured	12-23

Table 12.1: IP Global Parameters – Layer 3 Switches (Continued)

Parameter	Description	Default	See page...
IP unicast cache performance mode	The amount of available IP cache that is set aside for IP unicast entries. When the router caches unicast forwarding entries, the cached entries provide an optimal path through the router because the router CPU does not need to process the packets for forwarding. Once a packet is processed, the forwarding information is placed in the cache for reuse. Chassis devices provide an optional high-performance mode for allocating additional cache space for unicast forwarding entries. Use this option when the router is handling a very large number of unicast flows (source plus destination pairs) and you want to ensure that more flows can remain in the cache at one time.	High-performance mode enabled	12-80
IP load sharing	A Foundry feature that enables the router to balance traffic to a specific destination across multiple equal-cost paths. Load sharing uses a simple round-robin mechanism and is based on destination address. **Note**: Load sharing is sometimes called Equal Cost Multi Path (ECMP).	Enabled	12-66
IP load sharing aggregation	A feature on Chassis devices that increases the capacity of the load sharing cache by aggregating destination addresses into networks. When IP load sharing aggregation is enabled, each cache entry is an aggregate network for multiple destination hosts. If IP load sharing aggregation not enabled, the device creates a separate load sharing cache entry for each destination host address. **Note**: Load sharing aggregation is not available on Stackable devices. Stackable devices cache load sharing entries based on destination host addresses.	On Chassis devices, aggregated by network On Stackable devices, single host entries	12-77
Maximum IP load sharing paths	The maximum number of equal-cost paths across which the Layer 3 Switch is allowed to distribute traffic.	Four	12-79
CAM programming	Whether the device programs separate route entries into the CAM for individual route destinations or programs single aggregate entries for multiple destinations.	Separate entries are programmed for each destination	12-79
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	13-11 15-40 16-34

Table 12.1: IP Global Parameters – Layer 3 Switches (Continued)

Parameter	Description	Default	See page...
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	12-65
Static route	An IP route you place in the IP route table.	No entries	12-54
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.	12-41
IP option process	This parameter allows you to protect the device from attackes contained in the option field of an IP header. This parameter is available on the BigIron MG8 and NetIron 40G, this command is available in software release 02.2.01 and later.	Disabled	12-39

IP Interface Parameters – Layer 3 Switches

Table 12.2 lists the interface-level IP parameters for Layer 3 Switches.

Table 12.2: IP Interface Parameters – Layer 3 Switches

Parameter	Description	Default	See page...
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]	12-19
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	12-32

Table 12.2: 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	12-35
ARP age	Locally overrides the global setting. See Table 12.1 on page 12-10.	Ten minutes	12-45
Directed broadcast forwarding	Locally overrides the global setting. See Table 12.1 on page 12-10.	Disabled	12-50
ICMP Router Discovery Protocol (IRDP)	Locally overrides the global IRDP settings. See Table 12.1 on page 12-10.	Disabled	12-88
ICMP Redirect messages	Locally overrides the global setting. See Table 12.1 on page 12-10.	Enabled	12-54
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 (bootpc) must be enabled 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	12-96
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	12-92
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	12-93

Table 12.2: IP Interface Parameters – Layer 3 Switches (Continued)

Parameter	Description	Default	See page...
Egress priority merging	BigIron MG8 and NetIron 40G only. Egress priority merging, which merges internal priority (calculated on ingress) with the incoming VLAN tag priority at the egress on every port is disabled by default.	Disabled	12-95
Metric	A numeric cost the router adds to RIP routes learned on the interface. This parameter applies only to RIP routes.	1 (one)	13-4
IP state	The Internet Protocol, version 4	Enabled **Note**: You cannot disable IP.	n/a

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). NetIron Internet Backbone routers do not have a default IP address.

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 "Configuring IP Multicast Traffic Reduction" chapter in the *Foundry Switch and Router Installation and Basic Configuration Guide*.

IP Global Parameters – Layer 2 Switches

Table 12.3 lists the IP global parameters for Layer 2 Switches.

Table 12.3: 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.	12-104
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]	12-97
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	12-97
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

Table 12.3: IP Global Parameters – Layer 2 Switches (Continued)

Parameter	Description	Default	See page...
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	12-100
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	12-98
DNS default gateway addresses	A list of gateways attached to the router through which clients attached to the router can reach DNSs.	None configured	12-98
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	12-103

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). NetIron Internet Backbone routers do not have a default IP address.

Interface IP Parameters – Layer 2 Switches

Table 12.4 lists the interface-level IP parameters for Layer 2 Switches.

Table 12.4: 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	12-103

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 12-97.

Configuring IP Addresses

You can configure an IP address on the following types of Layer 3 Switch interfaces:

- Ethernet port
- Packet Over SONET (POS) 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 Stackable 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 "Displaying and Modifying System Parameter Default Settings" section of the "Configuring Basic Features" chapter of the *Foundry Switch and Router Installation and Basic Configuration Guide* or the *Foundry BigIron Mg8 Switch Installation and Basic Configuration Guide*.

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 12-104.

Assigning an IP Address to an Ethernet Port

To assign an IP address to an Ethernet port, use either of the following methods.

USING THE CLI

To assign an IP address to port 1/1, enter the following commands:

```
BigIron(config)# interface ethernet 1/1
BigIron(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:

```
BigIron(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.

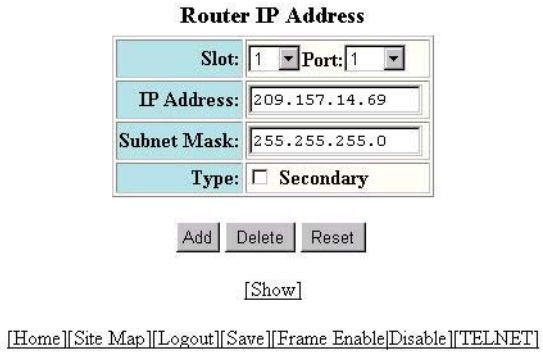
NOTE: When you configure more than one address in the same subnet, all but the first address are secondary addresses and do not form OSPF adjacencies.

USING THE WEB MANAGEMENT INTERFACE

To assign an IP address and mask to a router interface:

1. Log on to the Layer 3 Switch using a valid user name and password for read-write access. The System configuration dialog is displayed.

2. Select the IP Address link. The IP addresses already configured on the device are listed in a table. To add a new iP address link, select Add IP Address to display the following panel.



3. Select the port (and slot if applicable) on which you want to configure the address.

NOTE: This example shows the panel for configuring an address on a Layer 3 Switch. On a Layer 2 Switch, the IP address is global and applies to all the Layer 2 Switch's ports. Thus, you do not need to select a port.

4. Enter the IP address and network mask.
5. If the port already has an IP address configured, select the Secondary checkbox.
6. Click the Add button to save the change to the device's running-config file.
7. 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.

NOTE: You also can access the dialog for saving configuration changes by clicking on Command in the tree view, then clicking on Save to Flash.

Assigning an IP Address to a POS Port

Assigning an IP address to a POS port is similar to assigning an IP address to an Ethernet port.

NOTE: To configure other Layer 3 parameters for POS ports, see the "Using Packet Over SONET Modules" chapter in the *Foundry Switch and Router Installation and Basic Configuration Guide*.

USING THE CLI

To add an IP address to POS interface 2/1, enter the following commands:

```
BigIron(config)# interface pos 2/1
BigIron(config-posif-2/1)# ip address 209.157.22.26/24
BigIron(config-posif-2/1)# write memory
```

See the syntax description in "Assigning an IP Address to an Ethernet Port" on page 12-20.

USING THE WEB MANAGEMENT INTERFACE

See the procedure for Ethernet ports.

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 ona Stackable 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 16-13.

To add a loopback interface, use one of the following methods.

USING THE CLI

To add a loopback interface, enter commands such as those shown in the following example:

```
BigIron(config-bgp-router)# exit
BigIron(config)# int loopback 1
BigIron(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 12-20.

USING THE WEB MANAGEMENT INTERFACE

1. Log on to the Layer 3 Switch using a valid user name and password for read-write access. The System configuration panel is displayed.
2. Select the IP Address link to display a table listing the configured IP addresses.
3. Select the Loop Back link.

   NOTE: If the device already has loopback interfaces, a table listing the interfaces is displayed. Click the Modify button to the right of the row describing an interface to change its configuration, or click the Add Loop Back link to display the Router Loop Back configuration panel.

4. Select the loopback interface number from the Loopback field's pulldown menu. You can select from 1 – 8.
5. Select the status. The interface is enabled by default.
6. Click Add to add the new interface.
7. Click on Configure in the tree view to display the configuration options.
8. Click on IP to display the IP configuration options.
9. Select the Add IP Address link to display the Router IP Address panel.
10. Select the loopback interface from the Port field's pulldown menu. For example, to select loopback interface 1, select "lb1". (If you are configuring a Chassis device, you can have any slot number in the Slot field. Loopback interfaces are not associated with particular slots or physical ports.)
11. Enter the loopback interface's IP address in the IP Address field.
12. Enter the network mask in the Subnet Mask field.
13. Click the Add button to save the change to the device's running-config file.
14. 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.

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, IPX, or AppleTalk 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.

USING THE CLI

To add a virtual interface to a VLAN and configure an IP address on the interface, enter commands such as the following:

```
BigIron(config)# vlan 2 name IP-Subnet_1.1.2.0/24
BigIron(config-vlan-2)# untag e1 to 4
BigIron(config-vlan-2)# router-interface ve1
BigIron(config-vlan-2)# interface ve1
BigIron(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 12-20.

Deleting an IP Address

To delete an IP address, enter a command such as the following:

```
BigIron(config-if-1/1)# 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:

```
BigIron(config-if-1/1)# no ip address *
```

Syntax: no ip address <ip-addr> | *

Configuring Domain Name Server (DNS) Resolver

The Domain Name Server (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. On most Foundry devices, the feature lets you 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.

When a client performs a DNS query, all hosts within that domain can be recognized. After you define a domain name, the Foundry Layer 2 Switch or Layer 3 Switch automatically appends the appropriate domain to a host and forwards it to the DNS servers for resolution.

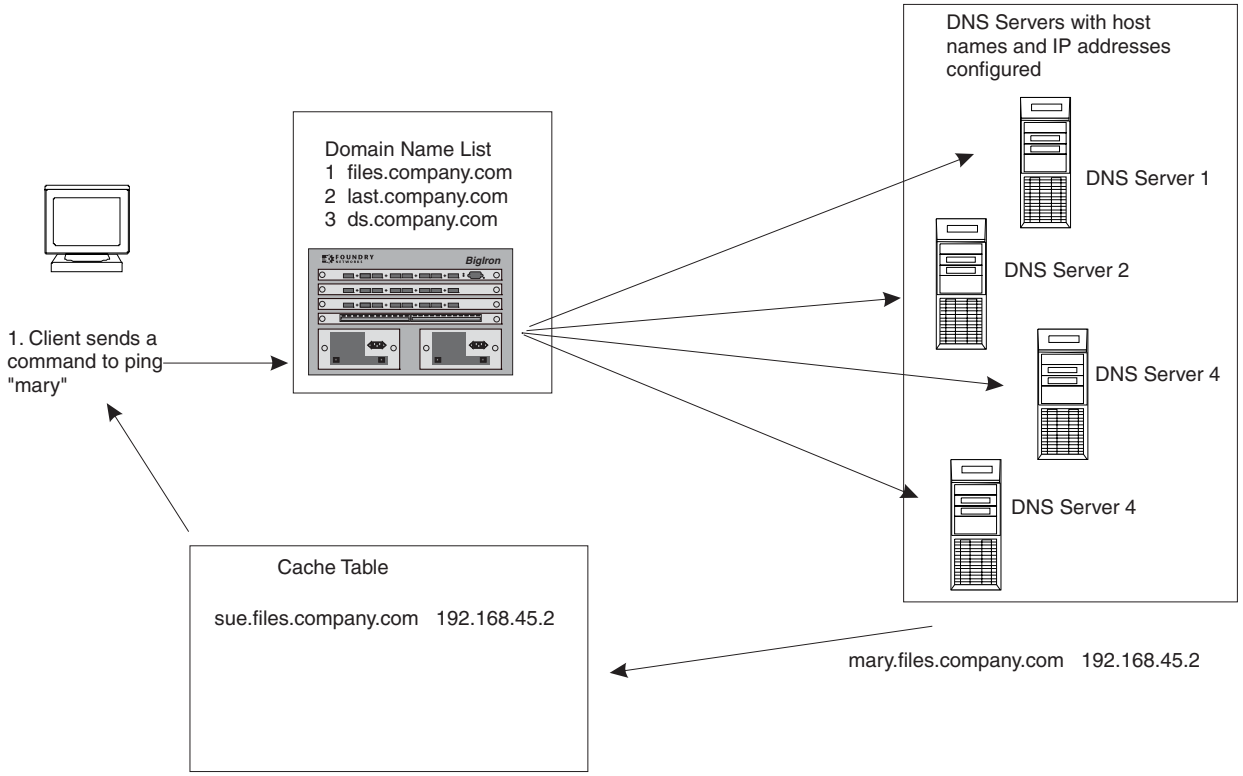
1.Foundry’s feature that allows routing between VLANs within the same device, without the need for external routers, is called Integrated Switch Routing (ISR). See the “Using Packet Over SONET Modules” chapter in the *Foundry Switch and Router Installation and Basic Configuration Guide*.

For example, if the domain “eng.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.eng.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 12.2.)

Figure 12.2 DNS resolution with one domain name

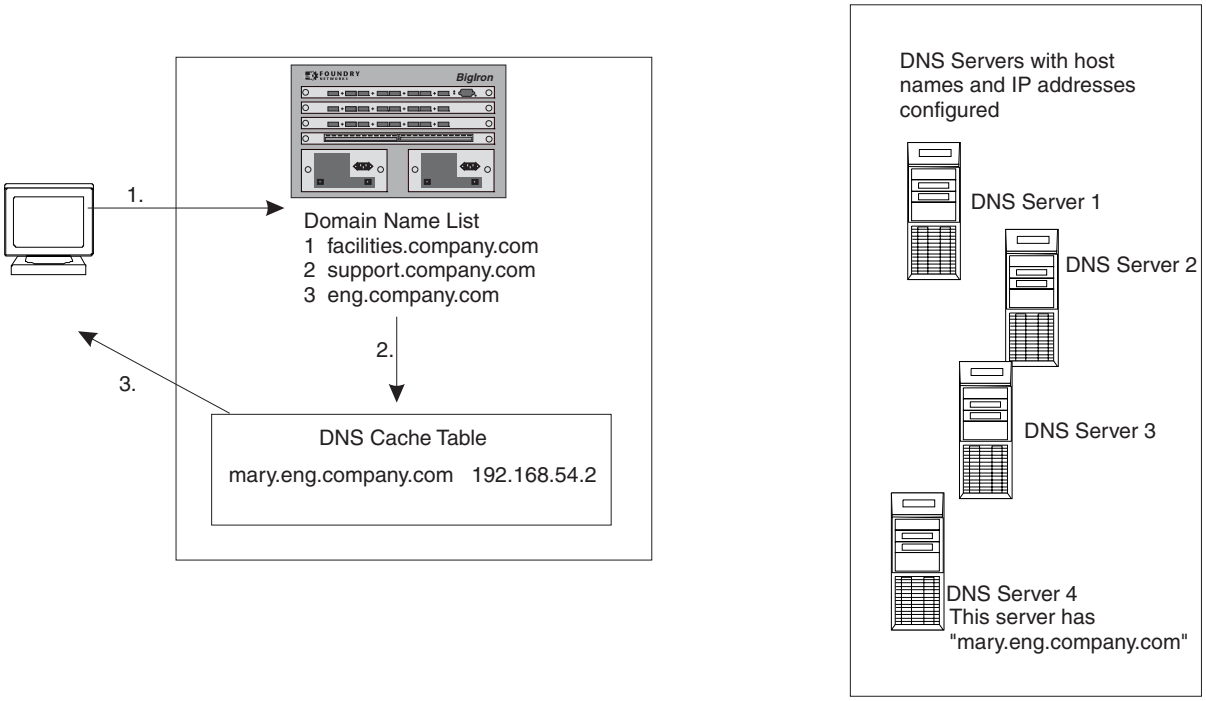


On devices running Enterprise software Release 08.0.00 and later, you can create a list of domain names that can be used to resolve host names. This list can have more than one domain name. When a client performs a DNS query all hosts within the domains on the list can be recognized and queries can be sent to any domain on the list.

Also, the Foundry device now contains a DNS cache table that contains a list of host names that have been resolved to their IP addresses. This DNS cache table allows DNS queries to be processed quickly. When a DNS query is made, the query can be sent to the DNS cache table. If a match is found, the DNS query is resolved. If no match is found, the DNS query is sent to the DNS server to be resolved before any action can be taken.

For example, in Figure 12.4, the client sends a ping to “mary”, which is in the eng.company.domain. The Foundry device appends the first domain name on the domain name list to “mary” and sends “mary.eng.company.com” to the DNS cache table. Since “mary.eng.company.com” is in the DNS cache table, the query is resolved quickly and the Foundry device returns the IP address of “mary”.

Figure 12.3 DNS resolution with host name in DNS Cache Table

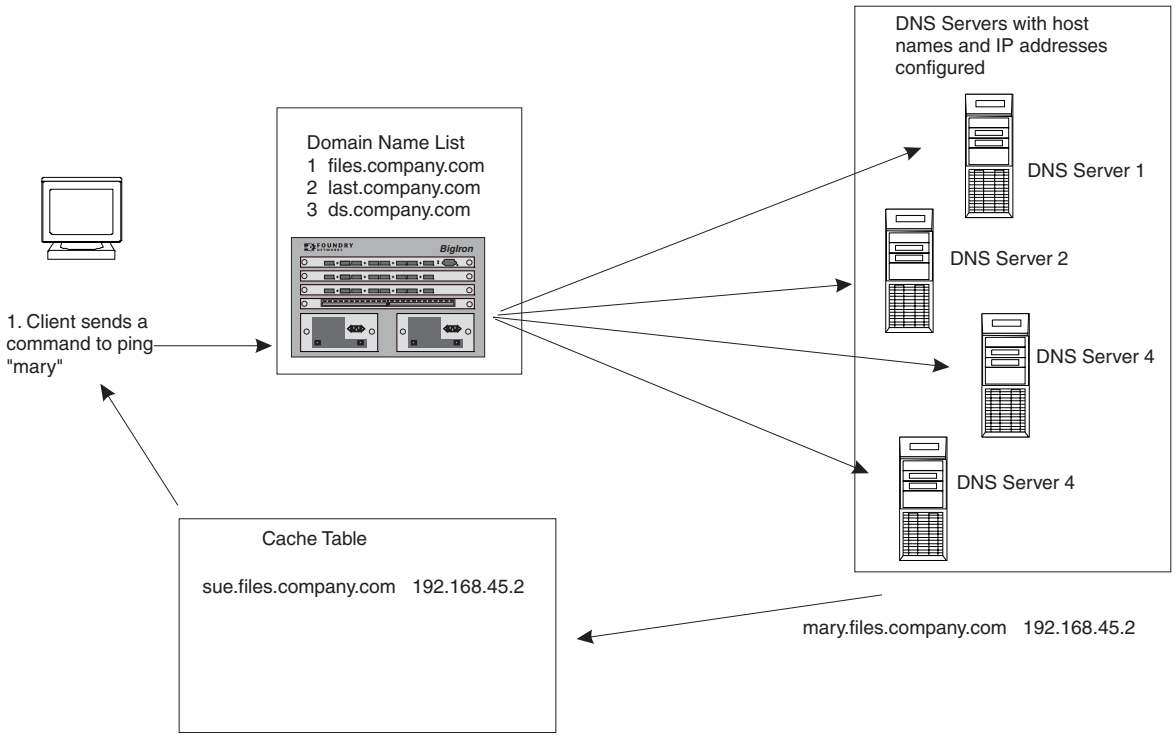


However, if “mary.eng.company.com” is not in the DNS cache table, as in Figure 12.4, the host name is resolved as follows:

1. A command to ping “mary” is entered at the client.
2. The Foundry device appends the first domain name to “mary” and sends the qualified host name “mary.facilities.company.com” to the DNS Cache table.
3. The DNS cache table does not have a “mary.facilities.company.com” entry, so it sends the host name to the DNS servers. Each DNS server is tried in sequential order.
4. Since none of the DNS servers have an entry to “mary.facilities.company.com”, the request is sent back to the Domain Name List. The next domain name is appended to “mary”.

   Step 2 through Step 4 are repeated until all the domains in the domain name list are tried. In Figure 12.4, “mary.eng.company.com” is found in DNS Server 4.
5. Since a match is found, the host name “mary.eng.company.com” and its IP address is added to the DNS cache table.
6. The host’s IP address is returned to the client. However, if no match is found, an “unknown host” message is returned to the client.

Figure 12.4 DNS resolution with host name not in DNS Cache Table



Over a period of time, there may be changes to the information in the DNS cache table. For example, a host's IP address can change, making the entries in the DNS cache table to be invalid. The Foundry device polls each entry in the DNS cache table to determine if the information in the DNS cache table is still valid. By default, the Foundry device sends a ping to the host every 1 minute. This polling interval can be changed.

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.

USING THE CLI

Suppose you want to define the domain name of newyork.com on a Layer 3 Switch and then define four possible default DNS gateway addresses. To do so, enter the following commands:

```
BigIron(config)# ip dns domain-name newyork.com
BigIron(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 THE WEB MANAGEMENT INTERFACE

To map a domain name server to multiple IP addresses:

1. Log on to the Layer 3 Switch using a valid user name and password for read-write access. The System configuration panel is displayed.
2. Click on the plus sign next to Configure in the tree view, then click on the plus sign next to IP, then select DNS to display the DNS panel.

3. Enter the domain name in the Domain Name field.
4. Enter an IP address for each device that will serve as a gateway to the domain name server.

NOTE: The first address entered will be the primary DNS gateway address. The other addresses will be used in chronological order, left to right, if the primary address is available.

5. Click the Apply button to save the change to the device's running-config file.
6. 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.

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 newyork.com domain is already defined on the Layer 3 Switch, you need to enter only the host name, NYC02, as noted below.

USING THE CLI

```
BigIron# 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. See the *Foundry Switch and Router Command Line Interface Reference* for information about the parameters.

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.

Defining a Domain Name

If you want to define only one domain to resolve host names, enter a command such as the following:

```
BigIron(config)# ip dns domain-name eng.company.com
```

Syntax: [no] ip dns domain-name <domain-name>

Enter the domain name for <domain-name>.

This command is available on all Foundry devices.

Defining a Domain List

If you want to use more than one domain name to resolve host names, you can create a list of domain names on Foundry devices running Enterprise software release 08.0.00 and later. For example, enter the commands such as the following:

```
BigIron(config)#ip dns domain-list facilities.company.com
BigIron(config)#ip dns domain-list support.company.com
BigIron(config)#ip dns domain-list eng.company.com
```



```
BigIron(config)#
```

The domain names are tried in the order you enter them

Syntax: [no] ip dns domain-list <domain-name>

Enter the full domain name for <domain-name>.

Use the **no** form of the command to remove a domain name.

Displaying the Domain Name List

To determine what domain names have been configured in the domain list of Foundry devices running Enterprise software release 08.0.00 and later, enter the following command:

```
BigIron(config)#show ip dns domain-list

1 facilities.company.com
2 support.company.com
3 eng.company.com
BigIron(config)#
```

Syntax: show ip dns domain-list

Defining DNS Servers

On Foundry devices running Enterprise software release 08.0.00 and later, you can configure Foundry device to recognize up to four DNS servers. The first entry serves as the primary (default) DNS server. If a query to the primary DNS server fails to be resolved after three attempts, the next DNS server is queried (also up to three times). This process continues for each defined DNS server until the query is resolved. The order in which the DNS servers are polled is the same as the order in which you enter them.

To define DNS servers, enter a command such as the following:

```
BigIron(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 the example above, the first DNS server entered becomes the primary DNS server and all others are secondary servers. Because DNS IP address 201.98.7.15 is the last DNS server listed, it is also the last DNS server consulted to resolve a query.

Verifying Domain Name or IP Address

On Foundry devices running Enterprise software release 08.0.00 and later, you can use the **ip domain-lookup** command to verify the host name for an IP address or the IP address for a host name. For example, if you have an IP address and you want to find out what host name it resolves to, enter the following command:

```
BigIron#ip domain-lookup 66.151.144.5

Host                                Flag     TTL/min Type Address
border2.pc0-0-bbnet1.sje.pnap.net   (TMP,OK) 720     IP   66.151.144.5
BigIron#
```

You can also enter the following:

```
BigIron#ip domain-lookup border2
Host                                Flag     TTL/min Type Address
border2.pc0-0-bbnet1.sje.pnap.net   (TMP,OK) 720     IP   66.151.144.5
BigIron#
```

Syntax: ip domain-loopkup <ip-address> | <host-name>

Enter an IP address to obtain the host name. Enter the host name to obtain the IP address.

The complete, qualified host name, along with its IP address and TTL value are displayed.

Adding Host Names to the DNS Cache Table

The entries in a DNS cache table are used to resolve host names to IP addresses on Foundry devices running Enterprise software release 08.0.00 and later. When a client initiates a DNS query, the Foundry device checks the DNS cache table to see if the host name can be resolved to any of the entries. If it a match is found, the query is resolved. If a match is not found, the DNS resolver sends the query to the DNS servers. If the name is resolved, the complete, qualified host name and its IP address is added to the DNS cache table and the hosts' IP address is returned to the client.

You can manually add entries to the DNS cache table if you know a host's complete, qualified name and its IP address. To add host names and their IP addresses to the DNS cache table, enter commands such as the following:

```
BigIron(config)#ip dns cache-entry www.foundrynet.com 63.236.63.244 720 dynamic-
cache-entry
```

Syntax: [no] ip dns cache-entry <host-name> <ip-address> <ttl-value> dynamic-cache entry | static-cache-entry

Enter the a complete, qualified name for <host-name>. For example, enter www.company.com or host.company.com.

Enter the IP address of the host. This must be the correct IP address for the host.

Enter a time to live (TTL) value for <ttl-value>. The TTL determines how many minutes host information stays in the DNS cache table if it has been dynamically added. Once the TTL value expires, the dynamically added host is removed from the table. If the host is added as a static host, the TTL value never changes and the entry does not expire unless it is manually removed from the table.

Enter **dynamic-cache-entry** if you want the host to be listed in the DNS cache table for the duration of the TTL value you entered. Once the TTL value expires, the domain is removed from the DNS cache table.

Enter **static-cache-entry** if you want the domain to remain in the DNS cache table until it is manually cleared.

Use the **no** form of the command to manually remove an entry from the DNS cache table; however, you must enter the entire entry to delete the entry. For example, you must enter:

```
BigIron(config)#no ip dns cache-entry www.foundrynet.com 63.236.63.244 720 dynamic-
cache-entry
```

Use the **clear ip dns cache-table** command to clear all the entries in the DNS cache table.

Clearing the DNS Cache Table

To clear the entire DNS cache table on Foundry devices running Enterprise software release 08.0.00 and later,, enter the following command:

```
BigIron#clear ip dns cache-table
```

Syntax: clear ip dns cache-table

Displaying the DNS Cache Table

To display what hosts are currently in the DNS cache table on Foundry devices running Enterprise software release 08.0.00 and later, enter the following command:

```
BigIron(config)#show ip dns cache-table

Host                                Flag     TTL/min Type Address
border2.pc0-0-bbnet1.sje.pnap.net   (TMP,OK) 720     IP   66.151.144.5
sl-internap-109-0.sprintlink.net    (TMP,OK) 1440    IP   144.223.242.86
sl-st21-sj-13-0.sprintlink.net      (TMP,OK) 1440    IP   144.232.20.59
sl-bb21-sj-12-0.sprintlink.net      (TMP,OK) 1440    IP   144.232.3.201
sl-bb24-sj-9-0.sprintlink.net       (TMP,OK) 1440    IP   144.232.20.181
sl-bb21-stk-9-0.sprintlink.net      (TMP,OK) 1440    IP   144.232.4.245
sl-gw27-stk-1-2.sprintlink.net      (TMP,OK) 4319    IP   144.228.145.69
core1.ge0-1-bbnet2.sfj.pnap.net     (TMP,OK) 719     IP   216.52.0.65
border10.s5-10.sfj.foundry-6.sfj.pna (TMP,OK) 355    IP   63.251.227.173
www.australia.com                   (TMP,OK) 14      IP   66.151.135.102
www.foundrynet.com                  (TMP,OK) 26      IP   63.236.63.244
mail.company.com                    (STA,OK) 26      IP   64.236.22.148
BigIron(config)#
```

Syntax: show ip dns cache-table

This Column...	Displays...
Host	The complete, qualified domain name of the host.
Flag	Indicates if the entry is dynamic or static and if the information for the domain is up to date: • TMP – Entry is dynamic • STA – Entry is static • OK – Information for the entry is up to date • EX – Information for the entry is no longer valid
TTL/min	If the entry is dynamic (TMP) this value shows how long the entry remains in the DNS cache table. If the entry is static (STA), it remains in the DNS cache table and never changes until it is manually removed or the DNS cache table is cleared, even if it shows a TTL/min value.
Type	Type of IP address stored for the entry.
Address	The IP address of the entry.

Defining the Polling Interval

The polling interval determines how often the Foundry device running Enterprise software release 08.0.00 pings a host in the DNS cache table to determine if the information for that host has changed. If the ping request is successful, an OK value is entered for the host in the DNS cache table. If the host has been added as a dynamic entry, it's current TTL value is also updated. If the ping request is unsuccessful (for example, if the hosts's IP address is no longer valid) an EX value is entered for the host.

To define a polling interval, enter a command such as the following:


```
BigIron(config)#ip dns poll-interval 7
```

Syntax: ip dns poll-interval <minutes>

Enter the polling interval in minutes. The default is 1 minute.

Displaying the Polling Interval

To display the current polling interval configured for a Foundry devices running Enterprise software release 08.0.00 and later, enter the following command:

```
BigIron(config)#show ip dns poll-time-interval
Current DNS polling interval is 7 minutes
BigIron(config)#
```

Syntax: [no] show ip dns poll-time-interval

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.

```
BigIron# 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. See the *Foundry Switch and Router Command Line Interface Reference* for information about the parameters.

After you enter the command, a message indicating that the DNS query is in process and the current DNS 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 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 (configurable on JetCore devices only) – 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. On JetCore devices, you can set the global MTU up to 14336 bytes.
  - Port MTU (configurable on devices with JetCore and IronCore modules) – A port's default MTU depends on the encapsulation type enabled on the port.

NOTE: On a JetCore or IronCore device, you can globally increase the MTU for traffic between ATM or POS ports and Ethernet ports to 1920 bytes.

NOTE: This section describes how to change the encapsulation type and MTU for Ethernet ports. To change these parameters and other packet parameters for ATM or POS ports, see the "Using Asynchronous Transfer Mode Modules" or "Using Packet Over SONET Modules" chapter in the *Foundry Switch and Router Installation and Basic Configuration Guide*.

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.

NOTE: POS and ATM interfaces use different encapsulation types. See the "Using Packet Over SONET Modules" or "Using Asynchronous Transfer Mode Modules" chapter in the *Foundry Switch and Router Installation and Basic Configuration Guide*.

To change the IP encapsulation type on a Layer 3 Switch port, use either of the following methods.

USING THE CLI

To change the IP encapsulation type on interface 1/5 to Ethernet SNAP, enter the following commands:

```
BigIron(config)# int e 1/5
BigIron(config-if-5)# ip encapsulation ethernet_snap
```

Syntax: ip encapsulation ethernet_snap | ethernet_ii

USING THE WEB MANAGEMENT INTERFACE

1. Log on to the Layer 3 Switch using a valid user name and password for read-write access. The System configuration panel is displayed.
2. Click on the plus sign next to Configure in the tree view to expand the list of configuration options.
3. Click on the plus sign next to IP in the tree view to expand the list of IP option links.
4. Click on the Interface link to display the interface table.
5. Click on the Modify button in the row for the port.
6. Select the encapsulation type from the Encapsulation pulldown menu.

7. Click the Add button to save the change to the device's running-config file.
8. To configure settings for another port, select the port (and slot, if applicable) and go to step 6.
9. 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.

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 an individual ports.

The default MTU is 1500 bytes for Ethernet II packets and 1492 for Ethernet SNAP packets.

On Chassis devices with JetCore modules that has 1 Gigabit and 10 Gigabit ports, you can configure an MTU up to 14336 bytes, on a global or individual interface basis.

On Stackable devices with JetCore modules, you can configure an MTU up to 1792 bytes on a global or individual interface basis.

On IronCore devices, the maximum supported MTU is 1500 bytes for Ethernet II packets and 1492 for Ethernet SNAP packets. You can configure a lower MTU on an individual port basis. You cannot configure a higher MTU.

On devices with IronCore modules, for traffic between ATM and Ethernet ports only, you can globally increase the MTU to 1920 bytes. This MTU increase applies only to traffic forwarded through the device between ATM or POS and Ethernet ports, and does not affect the MTU for traffic forwarded between or onto Ethernet ports.

On the BigIron MG8 and NetIron 40G, this command is available in software release 02.2.01 and later, you can configure IP MTU to be greater than 1500 bytes, although the default remains at 1500 bytes. The size of the MTU you can define on these devices or on its interfaces 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).

JetCore MTU Enhancements

Software release 07.6.03 and later contain the following enhancements to JetCore 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 JetCore 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) – JetCore

NOTE: This section applies to JetCore devices and modules only. If your Chassis device is managed by a JetCore module, this section also applies to 10 Gigabit Ethernet modules in the chassis.

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.

To globally enable jumbo support on all ports, enter commands such as the following:

```
BigIron(config)# default-mtu 14336
BigIron(config)# write memory
BigIron(config)# end
BigIron# reload
```

Syntax: [no] default-mtu <num>

The <num> parameter specifies the maximum number of bytes an Ethernet frame can have in order to be forwarded on a port, as folows:

- On JetCore stackable devices, you can specify from 64 – 7168 bytes, on a global or individual interface basis. The default is 1518 bytes. If the 802.1X authentication is used and 802.1X supplicant will be sending packet that is greater than 1500 MTU, then default-mtu must be set to 1700 bytes.
- On JetCore chassis devices, you can specify from 64 – 14336 bytes, on a global or individual interface basis. The default is 1518 bytes. If the 802.1X authentication is used and 802.1X supplicant will be sending packet that is greater than 1500 MTU, then default-mtu must be set to 1700 bytes.

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

When you install a newly configured management module, you must power up or reboot the device, then enter a **write memory** command, followed by a **reload** command. If you do not do this, jumbo packet support will not be enabled even if a **show mtu** display shows the configured MTU size.

Changing the Maximum Transmission Unit on an Individual Port – JetCore

To change the MTU on individual ports, enter commands such as the following:

```
BigIron(config)# interface ethernet 1/1 to 1/2
BigIron(config-mif-1/1-1/2)# mtu 14336
BigIron(config-mif-1/1-1/2)# write memory
BigIron(config-mif-1/1-1/2)# end
BigIron# reload
```

Syntax: [no] mtu <num>

These commands change the MTU on ports 1/1 and 1/2 to 14336. When you set the MTU on an individual port or group of ports, this setting overrides the global MTU setting.

NOTE: You must save the configuration change and then reload the software to place the jumbo support into effect.
When you install a newly configured management module, you must power up or reboot the device, then enter a **write memory** command, followed by a **reload** command. If you do not do this, jumbo packet support will not be enabled even if a **show mtu** display shows the configured MTU size.

Changing the Maximum Transmission Unit on an Individual Port – IronCore

NOTE: This section applies only to IronCore devices.

NOTE: For software releases 07.6.02 and later, see the section "Increasing the MTU for Traffic Between ATM and Ethernet" on page 12-36.

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

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 place the jumbo support into effect.

NOTE: If the 802.1X authentication is used and the 802.1X supplicant will be sending packet that is greater than 1500 MTU, then the following MTU on devices with IronCore modules using the **jumbo 1920** command. Refer to "Increasing the MTU for Traffic Between ATM and Ethernet" on page 12-36.

To change the MTU for a port, use either of the following methods.

USING THE CLI

To change the MTU for interface 1/5 to 1000, enter the following commands:

```
BigIron(config)# int e 1/5
BigIron(config-if-5)# ip mtu 1000
BigIron(config-if-5)# write memory
BigIron(config-if-5)# end
BigIron# reload
```

Syntax: [no] ip mtu <num>

The <num> parameter specifies the MTU. Ethernet II packets can hold IP packets from 572 – 1500 bytes long, although this limit is not applicable to BigIron MG8 and NetIron 40G devices running software release 02.2.01 and later. (See "Changing the Maximum Transmission Unit (MTU)" on page 12-33.) Ethernet SNAP packets can hold IP packets from 572 – 1492 bytes long.

The default MTU for Ethernet II packets is 1500. The default MTU for SNAP packets is 1492.

USING THE WEB MANAGEMENT INTERFACE

1. Log on to the Layer 3 Switch using a valid user name and password for read-write access. The System configuration panel is displayed.
2. Click on the plus sign next to Configure in the tree view to expand the list of configuration options.
3. Click on the plus sign next to IP in the tree view to expand the list of IP option links.
4. Click on the Interface link to display the interface table.
5. Click on the Modify button in the row for the port.
6. Enter an MTU value from 572 – 1492 if the interface is operating with Ethernet SNAP encapsulation. If the interface is operating with Ethernet II, enter a value from 572 – 1500.
7. Click the Add button to save the change to the device's running-config file.
8. To configure settings for another port, select the port (and slot, if applicable) and go to step 6.

9. 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.

Increasing the MTU for Traffic Between ATM and Ethernet

NOTE: This section applies to IronCore and JetCore devices, and applies only to traffic between ATM or POS and Ethernet ports on the device.

For traffic between ATM and Ethernet ports, you can change the MTU to 1920 bytes.

Starting with release 07.6.02, you can configure some Ethernet interfaces on an IronCore device to have an MTU of 1518 bytes and others to have an MTU of 1920 bytes. In this case, the fragmentation boundary for packets received from an ATM interface that need to be forwarded to an Ethernet interface is based on the MTU of the outbound Ethernet interface. For example, if the MTU on an Ethernet interface is 1518, and the ATM MTU is 9180, the device fragments the IP payload of a 9180-byte packet received on the ATM interface into seven fragments and forwards the fragments to the destination Ethernet port. If the MTU on another Ethernet interface is 1920, the device fragments the IP payload of a 9180-byte packet into only five fragments. If ATM receives a packet of up to 1902 bytes, it will not fragment the packet. If ATM receives a packet of more than 1902 bytes, then the device fragments the packet into chunks of 1872 bytes until the last fragment is sent.

Software releases prior to 07.6.02 supported an MTU of 1920 bytes, but it had to be applied globally to all interfaces on the device. In this case, the global MTU size determines the fragmentation boundary for packets received from an ATM interface that need to be forwarded to an Ethernet interface. For example, if the global MTU is 1500 and the ATM MTU is 9180, the device fragments a 9180-byte packet received on the ATM interface into seven fragments and forwards the fragments to a destination Ethernet port. If the global MTU is 1920, the device fragments a 9180-byte packet into only five fragments. If ATM receives a packet of up to 1920 bytes, it will not fragment the packet. If ATM receives a packet of more than 1920 bytes, then it fragments the packet into chunks of 1872 bytes until the last fragment is sent.

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 place the jumbo support into effect.

NOTE: This command does not affect the global MTU for traffic between Ethernet ports. The Ethernet MTU is still listed as 1492 or 1500 bytes.

NOTE: Regardless of the setting of the global MTU, the MTU for ATM PVCs is configurable up to 9180 bytes and is 4470 bytes by default.

NOTE: For VE traffic, the fragmentation boundary is based on the MTU of the physical outbound interface. No fragmentation is performed on Layer 2 frames. Layer 2 frames that exceed the MTU of the outbound interface are dropped.

To change the global MTU to 1920, enter the following commands:

```
BigIron(config)# jumbo1920
BigIron(config)# write memory
BigIron(config)# end
BigIron# reload
```

Syntax: [no] jumbo1920

You must save the configuration and reboot the device in order for the command to take effect.

NOTE: In releases prior to 07.6.02, you set the default MTU for Ethernet interfaces to 1920 bytes with the **jumbo** command. Starting in release 07.6.02, the **jumbo** command was renamed to **jumbo1920**.

NOTE: The **jumbo1920** command is intended for use on IronCore devices only. On JetCore devices, use the **default-mtu** command at the global CONFIG level and the **mtu** command at the interface level. See "JetCore MTU Enhancements" on page 12-33 for more information. In addition, you should not use the **default-mtu** command to enable 1920-byte MTU support on the device.

After the system has been rebooted, the global MTU for all Ethernet interfaces on the device is changed to 1920 bytes. In release 07.6.02 and later, you can optionally change the MTU of individual Ethernet interfaces to 1518 bytes. For example:

```
BigIron(config)# int e 3/11
BigIron(config-if-e100-3/11)# mtu 1518
```

Syntax: mtu <bytes>

At the interface level, you can set the MTU to either 1518 or 1920 bytes.

Path MTU Discovery (RFC 1191) Support

Starting in release 07.6.02, 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, including ATM sub-interfaces.

No Fragmentation for Jumbo Packets Sent to the CPU (Release 07.8.00 and Higher)

In releases prior to 07.8.00, JetCore devices support forwarding of Layer 2 and Layer 3 jumbo packets for transit traffic passing through the device (that is, forwarded in hardware). However, jumbo packets originated or received by the CPU are divided into fragments no larger than 1500 bytes each. These kinds of packets include jumbo-sized control packets for IP protocols.

Starting in release 07.8.00, you can configure the Foundry device to send unfragmented jumbo-sized control packets from the CPU. The Foundry device can send IP packets to the CPU that are up to the maximum MTU of the port on which the packet was received. This improves throughput and forwarding efficiency for jumbo-sized control packets.

When jumbo-sized control packets are not fragmented, routing protocols can exchange routing table information with neighboring routers using fewer packets. For example, OSPF can send packets at the MTU size during the negotiation phase of forming an adjacency. In addition, DVMRP control packets can include more routes in a single route update, resulting in a faster convergence of the DVMRP route table.

This performance enhancement is activated by default. In order for an interface to receive jumbo packets, you must configure a large enough MTU on the interface.

Specifying an MTU for IP Control Packets

By default, the MTU used for IP control packets is the configured MTU for the interface, minus 18 bytes. For example, if the configured MTU on the interface is 9018 bytes, the maximum MTU for IP control packets is 9000 bytes. (The other 18 bytes is used for 14 bytes of MAC header, and 4 bytes of CRC.)

You can optionally specify an alternate MTU for IP control packets. The MTU specified for IP control packets overrides the interface's configured MTU. For example, the following commands set the MTU for IP control packets on interface 3/11 to 4096 bytes:

```
BigIron(config)# interface e 3/11
BigIron(config-if-3/11)# ip jumbo-mtu 4096
```

Syntax: [no] ip jumbo-mtu <length>

In this example, the maximum MTU for IP control packets is 4078 bytes (4096 bytes minus 18 bytes).

You can specify a value that is between 1500 bytes and the MTU configured for the interface.

NOTE: To set an IP MTU smaller than 1500 bytes, use the **ip mtu** command instead of the **ip jumbo-mtu** command. The **ip mtu** command is intended for use only when you want to configure an MTU smaller than 1500 bytes.

Notes Regarding OSPF Adjacencies

By default, the Foundry device compares the MTU in an OSPF database description packet received from another OSPF router with the MTU on the Foundry device. The Foundry device allows formation of an adjacency with the other router only if the MTU in the other router's packet matches the MTU on the Foundry device.

The following applies when the **ip jumbo-mtu** command is configured:

- If the Foundry device is connected to another Foundry device, the MTU used is the MTU configured on the interface, minus 18 bytes. For example, if two devices each have an MTU of 9018 bytes, the OSPF adjacency is established with an MTU of 9000 bytes.
- If the Foundry device is connected to a Cisco router, add 18 bytes to the MTU setting. For example, if the MTU of the Cisco router is 9000 bytes, the MTU of the Foundry device should be set to 9018 bytes.
- If the Foundry device is connected to a Juniper router, add 4 bytes to the MTU setting. For example, if the MTU of the Juniper router is 9000 bytes, the MTU of the Foundry device should be set to 9004 bytes.
- If the two routers have an MTU mismatch, you can configure the **ip ospf mtu-ignore** command at the interface CONFIG level to disable the MTU negotiation.

Per-VLAN Forwarding of Jumbo Packets (Release 07.8.00 and Higher)

When you configure the MTU on a port, the port is capable of transmitting jumbo packets. 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 07.8.00, you can configure the device to forward jumbo packets based on the packets' VLAN membership. Jumbo-sized packets can be forwarded on one VLAN, while another VLAN can be restricted to forwarding standard-sized packets.

To activate per-VLAN forwarding on the Foundry device, enter the following command:

```
BigIron(config)# vlan-l3jumbo
```

Syntax: [no] vlan-l3jumbo

The following commands set the MTU on port 1/1 9018 bytes:

```
BigIron(config)# int e 1/1
BigIron(config-if-e1000-1/1)# mtu 9018
BigIron(config-if-e1000-1/1)# exit
```

The following commands set up two VLANs consisting of port 1/1, and enable jumbo packet forwarding for one of the VLANs.

```
BigIron(config)# vlan 10
BigIron(config-vlan-10)# tagged e 1/1
BigIron(config-vlan-10)# permit-l3jumbo
BigIron(config-vlan-10)# router-interface ve 10
BigIron(config-vlan-10)# exit

BigIron(config)# vlan 20
BigIron(config-vlan-20)# tagged e 1/1
BigIron(config-vlan-20)# router-interface ve 10
BigIron(config-vlan-20)# exit
```

Syntax: [no] permit-l3jumbo

The following commands create two virtual routing interfaces

```
BigIron(config)# int ve 10
BigIron(config-vif-10)# ip address 10.10.10.1/24
```



```
BigIron(config-vif-10)# exit

BigIron(config)# int ve 20
BigIron(config-vif-20)# ip address 20.20.20.1/24
BigIron(config-vif-20)# exit
```

In this sample configuration, packets forwarded out virtual interface ve 10 are forward as jumbo, while packets forwarded out virtual interface ve 20 are fragmented.

Notes:

- The 10/100 ports on JetCore modules do not support the jumbo MTU setting. If 10/100 ports are part of the virtual routing interface, the effective MTU is the lowest MTU of all the ports in the VLAN.
- This enhancement applies only to JetCore devices.

Enabling IP Option Attack Protection

NOTE: This feature is available on the BigIron MG8 and NetIron 40G running software release 02.2.01 and later.

An attack on the network could be accomplished using the options field of an IP packet header. For example, the source routing option makes it possible for the sender to specify a route to follow.

To protect against attacks contained in the option field, BigIron MG8 and NetIron 40G devices drop any IP packet that contains an option in its header, except for IGMP packets. IGMP packets are processes even if they contain IP options. If you want other packets that contain options in their headers to be processed, enter a command such as the following:

```
BigIron MG8(config)#ip ip-option-process
```

Syntax: ip ip-option-process

Setting Maximum Frame Size Per PPCR (Terathon Devices)

On Terathon devices, beginning with Terathon software release 02.2.00, when you set a maximum frame size, that maximum applies to all ports that are associated with the same packet processor (PPCR). Table 12.5 shows the ports of each Interface module.

Table 12.5: Ports available per PPCR

module type	Number of Packet Processors (PPCR)	Module Port Range Belonging to each PPCR PPCR 1	PPCR 2	PPCR 3	PPCR 4	PPCR 5	PPCR 6	PPCR 6	PPCR 8
2 x 10G	2	1	2	N/A	N/A	N/A	N/A	N/A	N/A
4 x 10G	4	1	2	3	4	N/A	N/A	N/A	N/A
8 x 10G	8	1	2	3	4	5	6	7	8
10 x 1G	1	1 - 10	N/A	N/A	N/A	N/A	N/A	N/A	N/A
20 x 1G	2	1 - 10	11 - 20	N/A	N/A	N/A	N/A	N/A	N/A
40 x 1G	4	1 - 10	11 - 20	21 -30	31 - 40	N/A	N/A	N/A	N/A
60 x 1G	3	1 - 20	21 - 40	41 - 60	N/A	N/A	N/A	N/A	N/A

To set a maximum frame size for all the ports attached to a PPCR, enter a command such as the following at the Interface Configuration level:

```
BigIron MG8(config)#interface ethernet 6/4
```



```
BigIron MG8(config-if-e1000-6/4)#max-frame-size 1500 bytes
BigIron MG8(config-if-e1000-6/4)#write memory
BigIron MG8(config-if-e1000-6/4)#exit
BigIron MG8(config)#reload
```

In this example the maximum frame size is applied to port 4 of a 40 x 1G Ethernet Interface module. That means that this maximum will apply to ports 1 to 10 on the interface module.

Syntax: max-frame-size <frame-size>

The <frame-size> variable specifies the maximum frame size for each port that is connected the same PPCR as described in Table 12.5. Values can be from 64 to 9212 bytes.

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.

USING THE CLI

To change the router ID, enter a command such as the following:

```
BigIron(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.

USING THE WEB MANAGEMENT INTERFACE

1. Log on to the Layer 3 Switch using a valid user name and password for read-write access. The System configuration panel is displayed.
2. Click on the plus sign next to Configure in the tree view to expand the list of configuration options.
3. Click on the plus sign next to IP in the tree view to expand the list of IP option links.
4. Click on the General link to display the IP configuration panel.
5. Edit the value in the Router ID field. Specify a valid IP address that is not in use on another device in the network.
6. Click the Apply button to save the change to the device's running-config file.
7. 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.

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 POS port 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.

USING THE CLI

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:

```
BigIron(config)# int loopback 2
BigIron(config-lbif-2)# ip address 10.0.0.2/24
BigIron(config-lbif-2)# exit
BigIron(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 <portnum> | pos <portnum> | loopback <num> | ve <num>

The <num> parameter is a loopback interface or virtual interface number. If you specify an Ethernet or POS 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.

```
BigIron(config)# interface ethernet 1/4
BigIron(config-if-1/4)# ip address 209.157.22.110/24
BigIron(config-if-1/4)# exit
BigIron(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:

```
BigIron(config)# int ve 1
BigIron(config-vif-1)# ip address 10.0.0.3/24
BigIron(config-vif-1)# exit
BigIron(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 <portnum> | pos <portnum> | loopback <num> | ve <num>

The <num> parameter is a loopback interface or virtual interface number. If you specify an Ethernet or POS 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:

```
BigIron(config)# int ve 1
BigIron(config-vif-1)# ip address 10.0.0.3/24
BigIron(config-vif-1)# exit
BigIron(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 <portnum> | pos <portnum> | loopback <num> | ve <num>

The <num> parameter is a loopback interface or virtual interface number. If you specify an Ethernet or POS port, the <portnum> is the port's number (including the slot number, if you are configuring a Chassis device).

USING THE WEB MANAGEMENT INTERFACE

You cannot configure a single source interface for Telnet, TACACS/TACACS+, or RADIUS using the Web management interface.

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 12-45.

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

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.

USING THE CLI

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:

```
BigIron(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.

USING THE WEB MANAGEMENT INTERFACE

You cannot configure ARP rate limiting using the Web management interface.

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 change the ARP age on a Layer 3 Switch, use either of the following methods.

USING THE CLI

To globally change the ARP aging parameter to 20 minutes, enter the following command:

```
BigIron(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:

```
BigIron(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.

USING THE WEB MANAGEMENT INTERFACE

1. Log on to the Layer 3 Switch using a valid user name and password for read-write access. The System configuration panel is displayed.
2. Click on the plus sign next to Configure in the tree view to display the list of configuration options.
3. Click on the plus sign next to IP to display the list of IP configuration options.
4. Select the General link to display the IP configuration panel.
5. Enter a value from 0 – 240 into the ARP Age field.
6. Click the Apply button to save the change to the device's running-config file.
7. 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.

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. The feature is not supported on Foundry Layer 2 Switches.

To enable Proxy ARP, use either of the following methods.

USING THE CLI

To enable IP proxy ARP, enter the following command:

```
BigIron(config)# ip proxy-arp
```

To again disable IP proxy ARP, enter the following command:

```
BigIron(config)# no ip proxy-arp
```

Syntax: [no] ip proxy-arp

USING THE WEB MANAGEMENT INTERFACE

1. Log on to the Layer 3 Switch using a valid user name and password for read-write access. The System configuration panel is displayed.
2. Click on the plus sign next to Configure in the tree view to expand the list of configuration options.
3. Click on the plus sign next to IP in the tree view to expand the list of IP option links.
4. Click on the General link to display the IP configuration panel.
5. Select the Enable or Disable radio button next to Proxy ARP.
6. Click the Apply button to save the change to the device's running-config file.
7. 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.

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 12-48.

To display the ARP cache and static ARP table, see the following:

- To display the ARP table, see "Displaying the ARP Cache" on page 12-113.
- To display the static ARP table, see "Displaying the Static ARP Table" on page 12-115.

To configure a static ARP entry, use either of the following methods.

USING THE CLI

To create a static ARP entry on a, enter a command such as the following:

```
BigIron(config)# arp 1 192.53.4.2 1245.7654.2348 e 1/2
```

Syntax: [no] arp <num> <ip-addr> <mac-addr> ethernet <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.

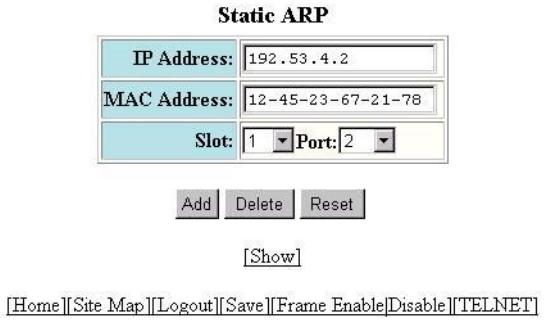
The <mac-addr> parameter specifies the MAC address of the entry.

The **ethernet** <portnum> command specifies the port number attached to the device that has the MAC address of the entry.

You can use the **show arp** and **show ip route** commands to determine if static routes are associated with a static ARP entry that binds multiple ports to the same destination.

USING THE WEB MANAGEMENT INTERFACE

1. Log on to the Layer 3 Switch using a valid user name and password for read-write access. The System configuration panel is displayed.
2. Click on the plus sign next to Configure in the tree view to expand the list of configuration options.
3. Click on the plus sign next to IP in the tree view to expand the list of IP option links.
4. Click on the General link to display the IP configuration panel.
5. Click the Static ARP link.
   - If the device does not have any static ARP entries, the Static ARP configuration panel is displayed, as shown in the following example.
   - If a static ARP entry is already configured and you are adding a new entry, click on the Add Static ARP link to display the Static ARP configuration panel, as shown in the following example.
   - If you are modifying an existing static ARP entry, click on the Modify button to the right of the row describing the entry to display the Static ARP configuration panel, as shown in the following example.



6. Enter the IP address. The address must be for a device that is directly connected to the Layer 3 Switch.
7. Enter the MAC address.
8. Select the port that the static ARP entry is to be assigned to from the pull down menu.
9. Click the Add button to save the change to the device's running-config file.
10. 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.

Configuring a Unicast Route With Multiple Outgoing Ports

Previous releases support sending traffic received from a VLAN or a Layer 3 routed interface to multiple Layer 2 addresses. In this kind of configuration, if a packet arrives within a VLAN that had a unicast IP destination address, but multicast MAC addresses, you can statically configure the multicast MAC address on two separate ports, and thus enable “flooding” on the two individual interfaces. If the packet arrives from another VLAN or Layer 3 routed interface, then the Foundry device takes the routed packet and applies the same multicast MAC “flooding” to the two statically defined ports.

Beginning with Enterprise software release 08.0.00, this functionality in now supported at Layer 3. The Foundry device can route incoming Layer 3 unicast IP packets to two or more statically defined outgoing ports. To configure the Foundry device to do this, you create a static ARP entry that specifies multiple ports for an IP address. The **arp** command has been enhanced to allow you to specify multiple output ports.

Notes

- There is no limitation on the type of MAC address (Layer 2 multicast, unicast) that can be used with this feature, nor is there a limitation on the number of outgoing ports.
- This feature can be used only in a Layer 3 topology; it cannot be used in a mixed Layer 2 and Layer 3 topology. The multiple outgoing ports must be routed interfaces. Sending unicast Layer 2 traffic to multiple outgoing ports (by statically configuring the same MAC address on two or more ports) has been supported in previous releases.
- The multiple outgoing ports must be Ethernet interfaces. The multiple outgoing ports cannot be virtual interfaces (VEs), ATM interfaces, POS interfaces, or trunk groups.
- This feature is not supported for the default route (0.0.0.0/0).

Creating Static ARP Entries with Multiple Outgoing Ports

To create a route that sends unicast Layer 3 traffic to multiple outgoing ports, you create a static ARP entry that specifies multiple ports for an IP address. The **arp** command has been enhanced to allow multiple output ports.

For example, to create a static ARP entry that has output ports of 1/21 and 1/22, enter the following command:

```
BigIron(config)# arp 1 20.20.20.2 0004.809e.2e15 multi-ports e 1/21 e 1/22
```

You can also specify the output ports as a range. For example:

```
BigIron(config)# arp 1 200.200.200.2 0004.809e.2e15 multi-ports e 1/21 to 1/22
```

Syntax: [no] arp <num> <ip-addr> <mac-addr> ethernet <portnum> |
multi-ports ethernet <portnum> [to <portnum>] [ethernet <portnum>]

Changing the Maximum Number of Entries the Static ARP Table Can Hold

Table 12.6 on page 12-49 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 “Displaying and Modifying System Parameter Default Settings” section of the “Configuring Basic Features” chapter of the *Foundry Switch and Router Installation and Basic Configuration Guide* or the *Foundry BigIron Mg8 Switch Installation and Basic Configuration Guide*.

USING THE CLI

To increase the maximum number of entries in the static ARP table you can configure on a BigIron Layer 3 Switch using a 512MB Management 4 module, enter commands such as the following at the global CONFIG level of the CLI:

```
BigIron(config)# system-max ip-static-arp 8000
BigIron(config)# write memory
```



```
BigIron(config)# end
BigIron# 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 following ranges, depending on the device you are configuring. The table below lists the default maximum and range of configurable maximums for static ARP table entries supported on each type of Foundry Layer 3 Switch.

Table 12.6: Static ARP Entry Support

Product	Default Maximum	Configurable Minimum	Configurable Maximum
NetIron Internet Backbone router with 512MB management module (Management 4 module)	2048	2048	10,000
BigIron with 512MB or 256MB Management 4 module	2048	2048	10,000
BigIron with 128MB management module (Management 2 or 3)	1024	1024	2048
BigIron with 32MB management module (Management 1 module)	512	512	1024
TurboIron/8 Layer 3 Switch with 32MB memory	512	512	1024
Stackable NetIron with 32MB memory	512	512	1024

USING THE WEB MANAGEMENT INTERFACE

To modify a table size using the Web management interface:

1. Log on to the Layer 3 Switch using a valid user name and password for read-write access. The System configuration panel is displayed.
2. Select the Max-Parameter link to display the Configure System Parameter Maximum Value table. This table lists the settings and valid ranges for all the configurable table sizes on the device.
3. Click the Modify button next to the ip-static-arp row.
4. Enter the new value for the cache size. The value you enter specifies the maximum number of entries the cache can hold.
5. Click Apply to save the changes to the device's running-config.
6. 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.
7. Click on the plus sign next to Command in the tree view to list the command options.
8. Select the Reload link and select Yes when the Web management interface asks you whether you really want to reload the software. Changes to cache and table sizes do not take effect until you reload the software.

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, use either of the following methods.

USING THE CLI

To modify the TTL threshold to 25, enter the following commands:

```
BigIron(config)# ip ttl 25
```

Syntax: ip ttl <1-255>

USING THE WEB MANAGEMENT INTERFACE

1. Log on to the Layer 3 Switch using a valid user name and password for read-write access. The System configuration panel is displayed.
2. Click on the plus sign next to Configure in the tree view to display the list of configuration options.
3. Click on the plus sign next to IP to display the list of IP configuration options.
4. Select the General link to display the IP configuration panel.
5. Enter a value from 1 – 255 into the TTL field.
6. Click the Apply button to save the change to the device's running-config file.
7. 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.

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, use either of the following methods.

USING THE CLI

```
BigIron(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:

```
BigIron(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:

```
BigIron(config)# interface ethernet 1/1
BigIron(config-if-1/1)# ip directed-broadcast
```

Syntax: [no] ip directed-broadcast

USING THE WEB MANAGEMENT INTERFACE

1. Log on to the Layer 3 Switch using a valid user name and password for read-write access. The System configuration panel is displayed.
2. Click on the plus sign next to Configure in the tree view to display the list of configuration options.
3. Click on the plus sign next to IP to display the list of IP configuration options.
4. Select the General link to display the IP configuration panel.
5. Select Enable or Disable next to Directed Broadcast Forward.
6. Click the Apply button to save the change to the device's running-config file.
7. 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.

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 12-52.

- 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.

USING THE CLI

To disable forwarding of IP source-routed packets, enter the following command:

```
BigIron(config)# no ip source-route
```

Syntax: [no] ip source-route

To re-enable forwarding of source-routed packets, enter the following command:

```
BigIron(config)# ip source-route
```

USING THE WEB MANAGEMENT INTERFACE

1. Log on to the Layer 3 Switch using a valid user name and password for read-write access. The System configuration panel is displayed.

2. Click on the plus sign next to Configure in the tree view to expand the list of configuration options.
3. Click on the plus sign next to IP in the tree view to expand the list of IP option links.
4. Click on the General link to display the IP configuration panel.
5. Select the Disable or Enable radio button next to Source Route.
6. Click the Apply button to save the change to the device's running-config file.
7. 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.

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 broadcasts, use either of the following methods.

USING THE CLI

To enable the Layer 3 Switch for zero-based IP subnet broadcasts in addition to ones-based IP subnet broadcasts, enter the following command.

```
BigIron(config)# ip broadcast-zero
```

Syntax: [no] ip broadcast-zero

USING THE WEB MANAGEMENT INTERFACE

You cannot enable zero-based IP subnet broadcasting using the Web management interface.

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. You can disable response to ping requests on a global basis using the following CLI method.

USING THE CLI

To disable response to broadcast ICMP echo packets (ping requests), enter the following command:

```
BigIron(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:

```
BigIron(config)# ip icmp echo broadcast-request
```

USING THE WEB MANAGEMENT INTERFACE

You cannot disable ICMP Echo replies using the Web management interface.

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.
- Network – The Foundry device cannot reach the network specified in the destination IP address of the packet.
- 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.

USING THE CLI

To disable all ICMP Unreachable messages, enter the following command:

```
BigIron(config)# no ip icmp unreachable
```

Syntax: [no] ip icmp unreachable [network | 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 **network** parameter disables ICMP Network Unreachable messages.
- The **host** parameter disables ICMP Host Unreachable messages.
- The **protocol** parameter disables ICMP Protocol Unreachable messages.

- 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 **port** parameter disables ICMP Port Unreachable messages.
- The **source-route-fail** parameter disables ICMP Unreachable (caused by Source-Route-Failure) messages.

To disable ICMP Host Unreachable messages and ICMP Network Unreachable messages but leave the other types of ICMP Unreachable messages enabled, enter the following commands instead of the command shown above:

```
BigIron(config)# no ip icmp unreachable host
BigIron(config)# no ip icmp unreachable network
```

If you have disabled all ICMP Unreachable message types but you want to re-enable certain types, you can do so entering commands such as the following:

```
BigIron(config)# ip icmp unreachable host
BigIron(config)# ip icmp unreachable network
```

The commands shown above re-enable ICMP Unreachable Host messages and ICMP Network Unreachable messages.

USING THE WEB MANAGEMENT INTERFACE

You cannot disable ICMP Destination Unreachable messages using the Web management interface.

Disabling ICMP Redirect Messages

You can disable or re-enable ICMP redirect messages. By default, a Foundry Layer 3 Switch sends an ICMP redirect message to the source of a misdirected packet in addition to forwarding the packet to the appropriate router. You can disable ICMP redirect messages on a global basis or on an individual port basis.

NOTE: The device forwards misdirected traffic to the appropriate router, even if you disable the redirect messages.

To disable ICMP redirect messages globally, enter the following command at the global CONFIG level of the CLI:

```
BigIron(config)# no ip icmp redirects
```

Syntax: [no] ip icmp redirects

To disable ICMP redirect messages on a specific interface, enter the following command at the configuration level for the interface:

```
BigIron(config)# int e 3/11
BigIron(config-if-e100-3/11)# no ip redirect
```

Syntax: [no] ip redirect

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 12-65.

- 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 or POS 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 12-66.
- 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 of 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 12-59
- “Configuring Standard Static IP Routes and Interface or Null Static Routes to the Same Destination” on page 12-61

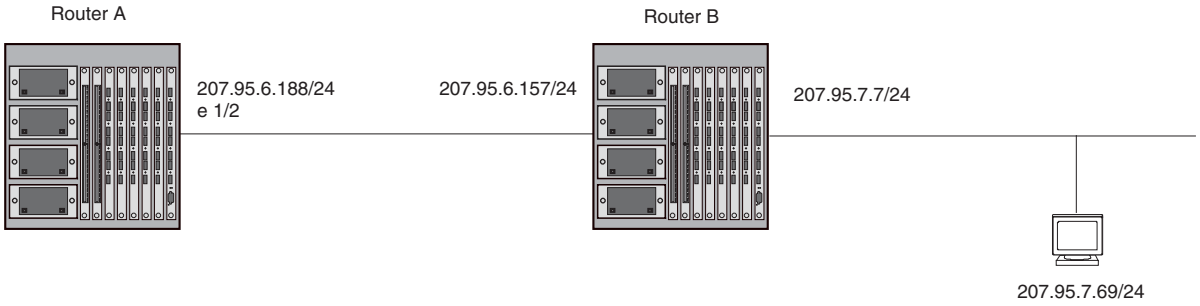
Static Route States Follow Port States

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 12.5 shows an example of a network containing a static route. The static route is configured on Router A, as shown in the CLI example following the figure.

Figure 12.5 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.

```
BigIron(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, Router 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. Router 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, use either of the following methods.

USING THE CLI

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:

```
BigIron(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.

```
BigIron(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.

```
BigIron(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 POS port 2/2 as its next hop.

```
BigIron(config)# ip route 192.128.2.73 255.255.255.0 pos 2/2
```

Syntax: ip route <dest-ip-addr> <dest-mask>
<next-hop-ip-addr> |
ethernet <portnum> | pos <portnum> | ve <num>
[<metric>] [distance <num>]

or

Syntax: ip route <dest-ip-addr>/<mask-bits>
<next-hop-ip-addr> |
ethernet <portnum> | pos <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 or POS 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 16-38 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.

USING THE WEB MANAGEMENT INTERFACE

1. Log on to the Layer 3 Switch using a valid user name and password for read-write access. The System configuration panel is displayed.
2. Click on the plus sign next to Configure in the tree view to expand the list of configuration options.
3. Click on the plus sign next to IP in the tree view to expand the list of IP option links.
4. Click on the General link to display the IP configuration panel.

5. Click the Static Route link.
   - If the device does not have any IP static routes, the Static Route configuration panel is displayed.
   - If a static route is already configured and you are adding a new route, click on the Add Static Route link to display the Static Route configuration panel.
   - If you are modifying an existing static route, click on the Modify button to the right of the row describing the static route to display the Static Route configuration panel.
6. Enter the network address for the route in the Network field.
7. Enter the network mask in the Mask field.
8. Select the next-hop type. You can select one of the following:
   - Address – The next-hop is the IP address of a gateway router.
   - Interface – The next hop is a port or virtual interface on the Layer 3 Switch.
9. Enter the next-hop IP address (if you selected the Address method) or select the interface (if you selected the Interface method).
   - Address – Enter the IP address of the next-hop gateway in the Next Hop (by Address) field.
   - Interface – Select the port, loopback interface, or virtual interface from the Next Hop (by Interface) field's pulldown menu(s). Loopback interfaces and virtual interfaces are listed in the Port pulldown menu, not in the Slot pulldown menu. To select a loopback interface or a virtual interface on a Chassis device, ignore the Slot pulldown menu and select the interface from the Port pulldown menu.
10. Optionally change the metric by editing the value in the Metric field. You can specify 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.

11. Optionally change the administrative distance by editing the value in the Distance field. 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.
12. Click the Add button to save the change to the device's running-config file.
13. Repeat steps 8 – 12 for each static route to the same destination.
14. 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.

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.

```
BigIron(config)# ip route 209.157.22.0 255.255.255.0 null0
BigIron(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.

Dropping Traffic Sent to the Null0 Interface In Hardware

In releases prior to 07.7.00, traffic sent to the null0 interface was done in software; that is, by sending the traffic to the CPU. Starting in release 07.7.00, this is now done in hardware; that is, by programming the CAM to discard traffic sent to the null0 interface. This improves forwarding efficiency and reduces the burden on the Foundry device's CPU.

Hardware dropping for IP traffic sent to the null0 interface is supported on devices with IronCore and JetCore, and on NPA OC-48 modules. For POS and ATM modules, traffic sent to the null0 interface is dropped by the module's CPU.

Note that this enhancement applies only when you are configuring a null static route. On IronCore devices, when you configure ACLs and route-maps with Policy-Based Routing (PBR) to send traffic to the null0 interface, the traffic is still dropped in software.

This feature applies to traffic not using the default IP route (0.0.0.0/0). If you define a null static route to drop traffic sent to the default IP route address, the traffic is still dropped by software. You can optionally configure the Foundry device to drop traffic sent to the default IP route address in hardware. To do this, enter the following commands:

```
BigIron(config)# ip route 0.0.0.0 0.0.0.0 null0
BigIron(config)# ip hw-drop-on-def-route
```

Syntax: [no] ip hw-drop-on-def-route

Configuring the the Foundry device to drop traffic sent to the default IP route address in hardware causes the device to program 32-bit host CAM entries for each destination address using the default route, which could consume the CAM space. To prevent this from happening, you can enable the CAM Default Route Aggregation feature. To do this, enter the following command:

```
BigIron(config)# ip dr-aggregate
```

Syntax: ip dr-aggregate

See "CAM Default Route Aggregation" on page 12-85 for more information.

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 12-66.

- 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 16-38.

The steps for configuring the static routes are the same as described in the previous section. The following sections provide examples.

USING THE CLI

To configure multiple static IP routes, enter commands such as the following.

```
BigIron(config)# ip route 192.128.2.69 255.255.255.0 209.157.22.1
BigIron(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.

```
BigIron(config)# ip route 192.128.2.69 255.255.255.0 209.157.22.1
BigIron(config)# ip route 192.128.2.69 255.255.255.0 192.111.10.1 2
BigIron(config)# ip route 192.128.2.69 255.255.255.0 201.1.1.1 3
```

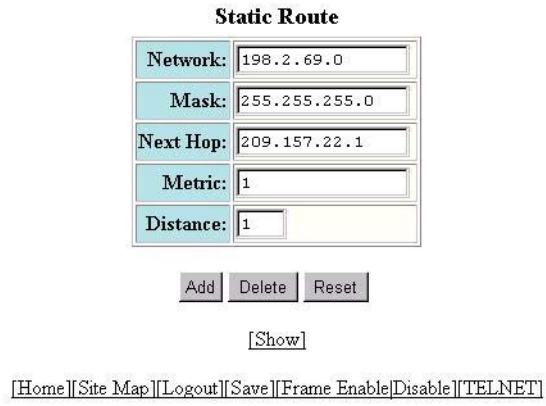
In this example, each static route has a different metric. The metric is not specified for the first route, so the default (1) is used. A metric is specified for the second and third static IP routes. The second route has a metric of two and the third route has a metric of 3. Thus, the second route is 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 12-56.

USING THE WEB MANAGEMENT INTERFACE

1. Log on to the Layer 3 Switch using a valid user name and password for read-write access. The System configuration panel is displayed.
2. Click on the plus sign next to Configure in the tree view to expand the list of configuration options.
3. Click on the plus sign next to IP in the tree view to expand the list of IP option links.
4. Click on the General link to display the IP configuration panel.
5. Click the Static Route link.
   - If the device does not have any IP static routes, the Static Route configuration panel is displayed, as shown in the following example.
   - If a static route is already configured and you are adding a new route, click on the Add Static Route link to display the Static Route configuration panel, as shown in the following example.
   - If you are modifying an existing static route, click on the Modify button to the right of the row describing

the static route to display the Static Route configuration panel, as shown in the following example.



6. Enter the network address for the route in the Network field.
7. Enter the network mask in the Mask field.
8. Enter the IP address of the next hop gateway in the Next Hop field.
9. Optionally change the metric by editing the value in the Metric field. You can specify 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.

---

10. Optionally change the administrative distance by editing the value in the Distance field. 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.
11. Click the Add button to save the change to the device's running-config file.
12. Repeat steps 8 – 11 for each static route to the same destination.
13. 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.

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 12.6 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 12.6 Standard and null static routes to the same destination network

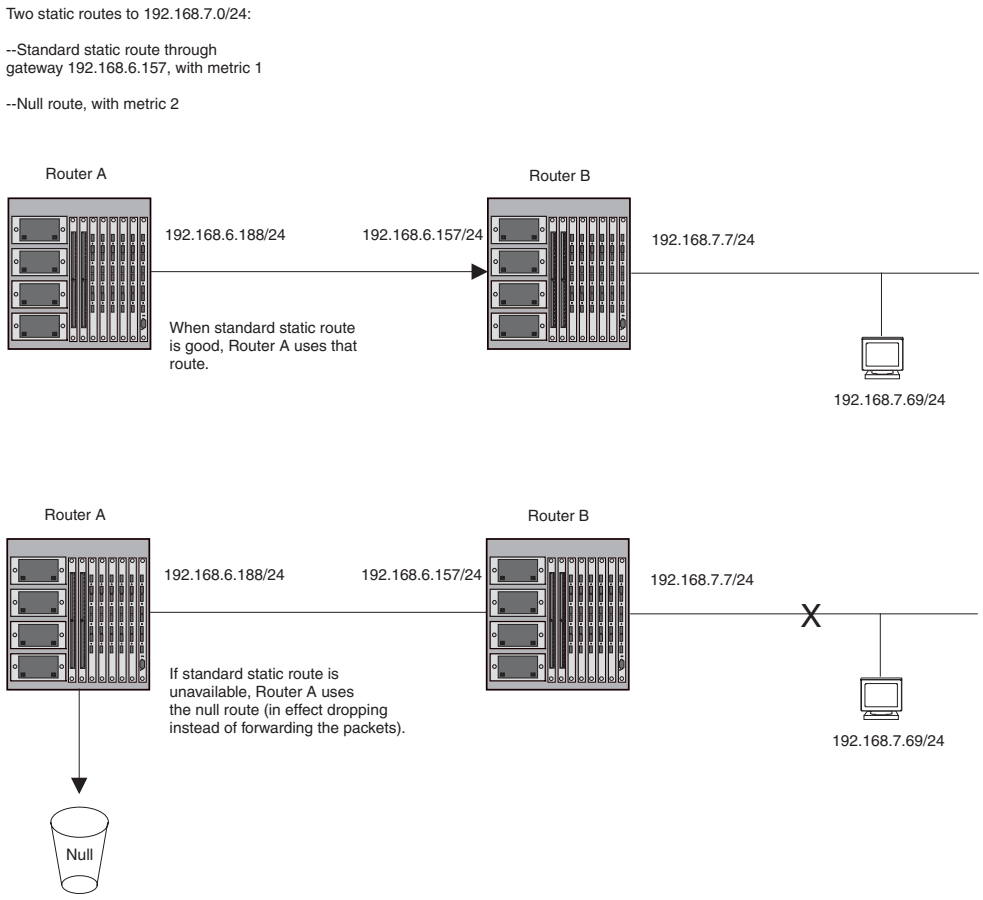
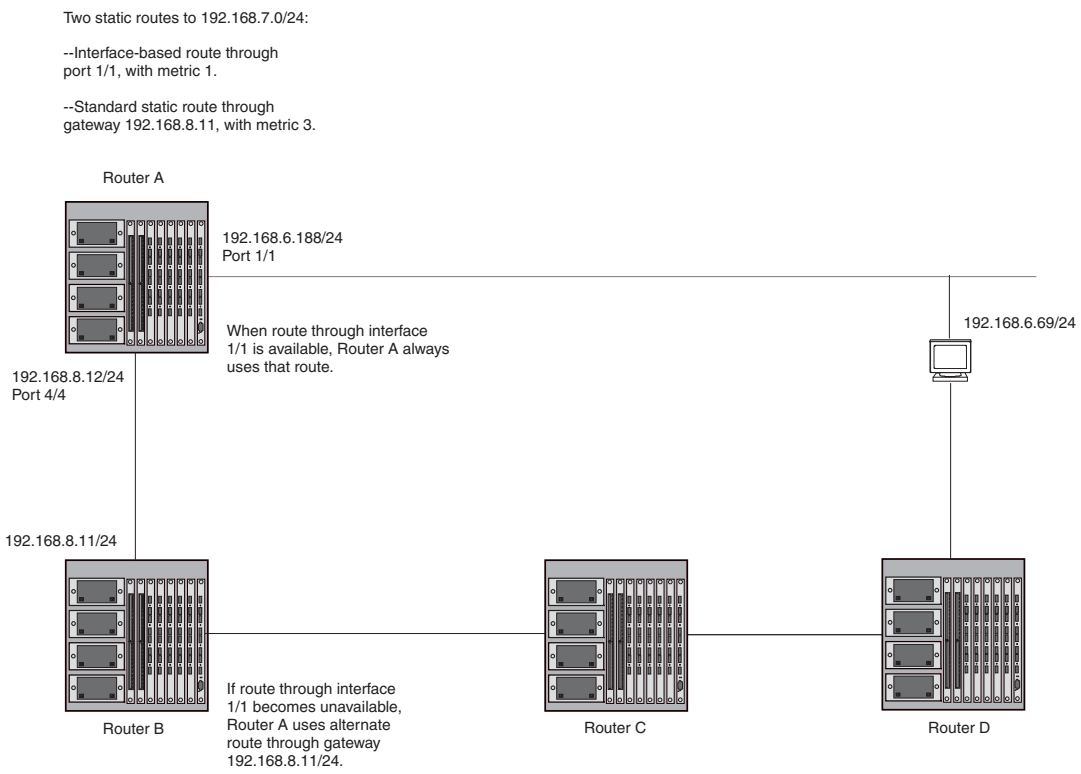


Figure 12.7 shows another example of two static routes. In this example, a standard static route and an interface-based static route are configured for destination network 192.168.6.0/24. The interface-based static route has a lower metric than the standard static route. As a result, the Layer 3 Switch 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 12.7 Standard and interface routes to the same destination network



To configure the multiple static routes of different types to the same destination, use either of the following methods.

USING THE CLI

To configure a standard static IP route and a null route to the same network as shown in Figure 12.6 on page 12-62, enter commands such as the following:

```
BigIron(config)# ip route 192.168.7.0/24 192.168.6.157/24 1
BigIron(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 12-56.

To configure a standard static route and an interface-based route to the same destination, enter commands such as the following:

```
BigIron(config)# ip route 192.168.6.0/24 ethernet 1/1 1
BigIron(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.

USING THE WEB MANAGEMENT INTERFACE

1. Log on to the Layer 3 Switch using a valid user name and password for read-write access. The System configuration panel is displayed.
2. Click on the plus sign next to Configure in the tree view to expand the list of configuration options.
3. Click on the plus sign next to IP in the tree view to expand the list of IP option links.
4. Click on the General link to display the IP configuration panel.
5. Click the Static Route link.
   - If the device does not have any IP static routes, the Static Route configuration panel is displayed.
   - If a static route is already configured and you are adding a new route, click on the Add Static Route link to display the Static Route configuration panel.
   - If you are modifying an existing static route, click on the Modify button to the right of the row describing the static route to display the Static Route configuration panel.
6. Enter the network address for the route in the Network field.
7. Enter the network mask in the Mask field.
8. Select the next-hop type. You can select one of the following:
   - Address – The next-hop is the IP address of a gateway router.
   - Interface – The next hop is a port, loopback interface, or virtual interface on the Layer 3 Switch.
9. Enter the next-hop IP address (if you selected the Address method) or select the interface (if you selected the Interface method).
   - Address – Enter the IP address of the next-hop gateway in the Next Hop (by Address) field.
   - Interface – Select the port, loopback interface, or virtual interface from the Next Hop (by Interface) field's pulldown menu(s). Loopback interfaces and virtual interfaces are listed in the Port pulldown menu, not in the Slot pulldown menu. To select a loopback interface or a virtual interface on a Chassis device, ignore the Slot pulldown menu and select the interface from the Port pulldown menu.

   NOTE: You cannot configure a null IP static route using the Web management interface.

10. Optionally change the metric by editing the value in the Metric field. You can specify 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.

11. Optionally change the administrative distance by editing the value in the Distance field. 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.
12. Click the Add button to save the change to the device's running-config file.
13. Repeat steps 8 – 12 for each static route to the same destination.
14. 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.

Adding a Tag to a Static Route

Static routes can be configured with a tag value, which can be used to color routes and filter routes during a redistribution process. When tagged static routes are redistributed to OSPF or to a protocol that can carry tag information, they are redistributed with their tag values.

To add a tag value to a static route, enter commands such as the following:


```
BigIron MG8(config)#ip route 192.122.12.1 255.255.255.0 192.122.1.1 tag 20
```

Syntax: ip route <dest-ip-addr> <dest-mask> | <dest-ip-addr>/<dest-mask> <next-hop-ip-address> tag <value>

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. You can enter multiple static routes for the same destination for load balancing or redundancy.

The <next-hop-ip-address> is the IP address of the next-hop router (gateway) for the route. In addition, the <next-hop-ip-address> can also be a virtual routing interface (for example. ve 100), or a physical port (for example, ethernet 1/1) that is connected to the next hop router.

Enter 0 – 4294967295 for **tag** <value>. The default is 0, meaning no tag.

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

To configure a default network route, use one of the following methods. You can configure up to four default network routes.

USING THE CLI

To configure a default network route, enter commands such as the following:

```
BigIron(config)# ip default-network 209.157.22.0
BigIron(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:

```
BigIron(config)# 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.

USING THE WEB MANAGEMENT INTERFACE

You cannot configure a default network route using the Web management interface. In addition, the IP route table display in the Web management interface does not indicate routes that are candidate default network routes. The routes are listed but are not flagged with an asterisk.

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]

IP load sharing is based on the destination address of the traffic. Chassis Layer 3 Switches support load sharing based on individual host addresses or on network addresses. Stackable Layer 3 Switches support load sharing based on host addresses.

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 not based on source routing, only on next-hop 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.

1.IP load sharing is also called "Equal-Cost Multi-Path (ECMP)" load sharing or just "ECMP"

NOTE: Foundry devices also perform load sharing among the ports in aggregate links. See the "Trunk Group Load Sharing" section in the "Configuring Trunk Groups and Dynamic Link Aggregation" chapter of the *Foundry Switch and Router Installation and Basic Configuration Guide*.

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
- 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 12-59.
- 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 12.7 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 12.7: Default Load Sharing Parameters for Route Sources

Route Source	Default Number of Paths	Default Maximum Number of Paths	Maximum Number of Paths	See...
Static IP route	4	4	8[1]	12-79
RIP	4	[4]	8[1]	12-79
OSPF	4	4	8[1]	12-79
BGP4	1	4	8[1]	16-29

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 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.

Foundry Layer 3 Switches support the following IP load sharing methods:

- Host-based – The Layer 3 Switch uses a simple round-robin mechanism to distribute traffic across the equal-cost paths based on destination host IP address. This is the only method supported by Stackable Layer 3 Switches and also is supported on Chassis Layer 3 Switches.
- Network-based – The Layer 3 Switch 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 method is available only on Chassis Layer 3 Switches and is the default.

NOTE: FastIron Edge Switches use host-based IP load sharing. This is the only type of IP load sharing supported onFastIron Edge Switches.

In addition, on Chassis Layer 3 Switches you can use network-based load sharing as the default while configuring host-based load sharing for specific destination networks. When you configure host-based load sharing for a specific destination network, the Layer 3 Switch distributes traffic to hosts on the network evenly across the available paths. For other networks, the Layer 3 Switch uses a single path for all traffic to hosts on a given network.

NOTE: Regardless of the method of load sharing that is enabled, the Layer 3 Switch always load shares paths for default routes and the network default route based on destination host address.

NOTE: The VM1 uses hash-based load-balancing of equal-cost entries instead of host-based or network-based load balancing. A hash value is calculated based on the source and destination IP addresses. Each of the paths to a given destination is associated with one of the possible hash values, and the traffic flow is assigned to a path based on its calculated hash value. Hash-based load sharing applies to traffic forwarded by software, not to traffic forwarded by hardware. Normally, traffic is forwarded in software when you configure a CPU-based feature such as ACLs, rate limiting, or NetFlow. Traffic also is forwarded by software if the CAM (used for hardware forwarding) becomes full.

Path Redundancy

If a path to a given destination becomes unavailable, the Layer 3 Switch provides redundancy by using another available equal-cost path to the destination, as described in the following sections.

NOTE: The following sections do not apply to the VM1. See the note above.

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:

- For host-based IP load sharing, the next load-balancing cache entry uses the first path to the destination. The first path is the path that entered the IP route table first. “Host-Based IP Load Sharing” on page 12-69 describes the host-based load-sharing mechanism.
- For network-based IP load sharing, the next load-balancing cache entry uses the next available path is then calculated based on the current number of paths and the maximum number of paths allowed. “Network-Based IP Load Sharing” on page 12-71 describes the network-based load-sharing mechanism.

Host-Based IP Load Sharing

The host-based load sharing method uses a simple round-robin mechanism to select an equal-cost path for traffic to a destination host. When the Layer 3 Switch receives traffic for a destination host and the IP route table has

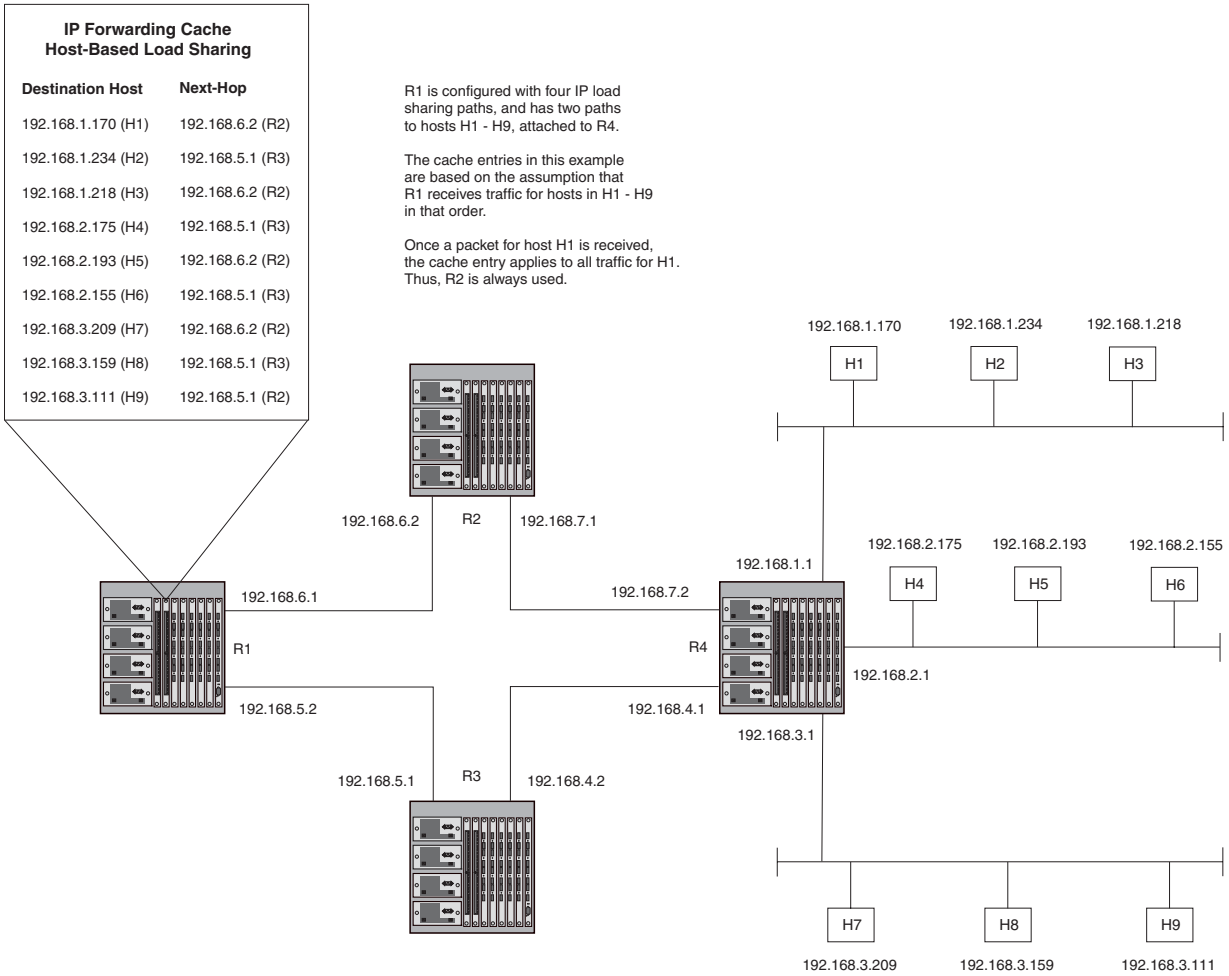
multiple equal-cost paths to the host, the Layer 3 Switch checks the IP forwarding cache for a forwarding entry to the destination.

- If the IP forwarding cache contains a forwarding entry for the destination, the device uses the entry to forward the traffic.
- If the IP forwarding cache does not contain a forwarding entry for the destination, the software selects the next path in the rotation (the path after the one the software used for the previous load sharing selection). The software then creates an IP forwarding cache entry that associates the destination host IP address with the selected path (next-hop IP address).

A cache entry for host-based IP load sharing has an age time of ten minutes. If a cache entry is not used before the age time expires, the device deletes the cache entry. The age time for IP load sharing cache entries is not configurable.

Figure 12.8 shows an example of host-based IP load sharing. In this example, the Layer 3 Switch has two equal-cost paths to hosts H1 – H9. For simplicity, this example assumes that the Layer 3 Switch does not have any entries in its IP forwarding cache to begin with, and receives traffic for the destination hosts (H1 – H9) in ascending numerical order, beginning with H1 and ending with H9.

Figure 12.8 Host-based IP load sharing – basic example

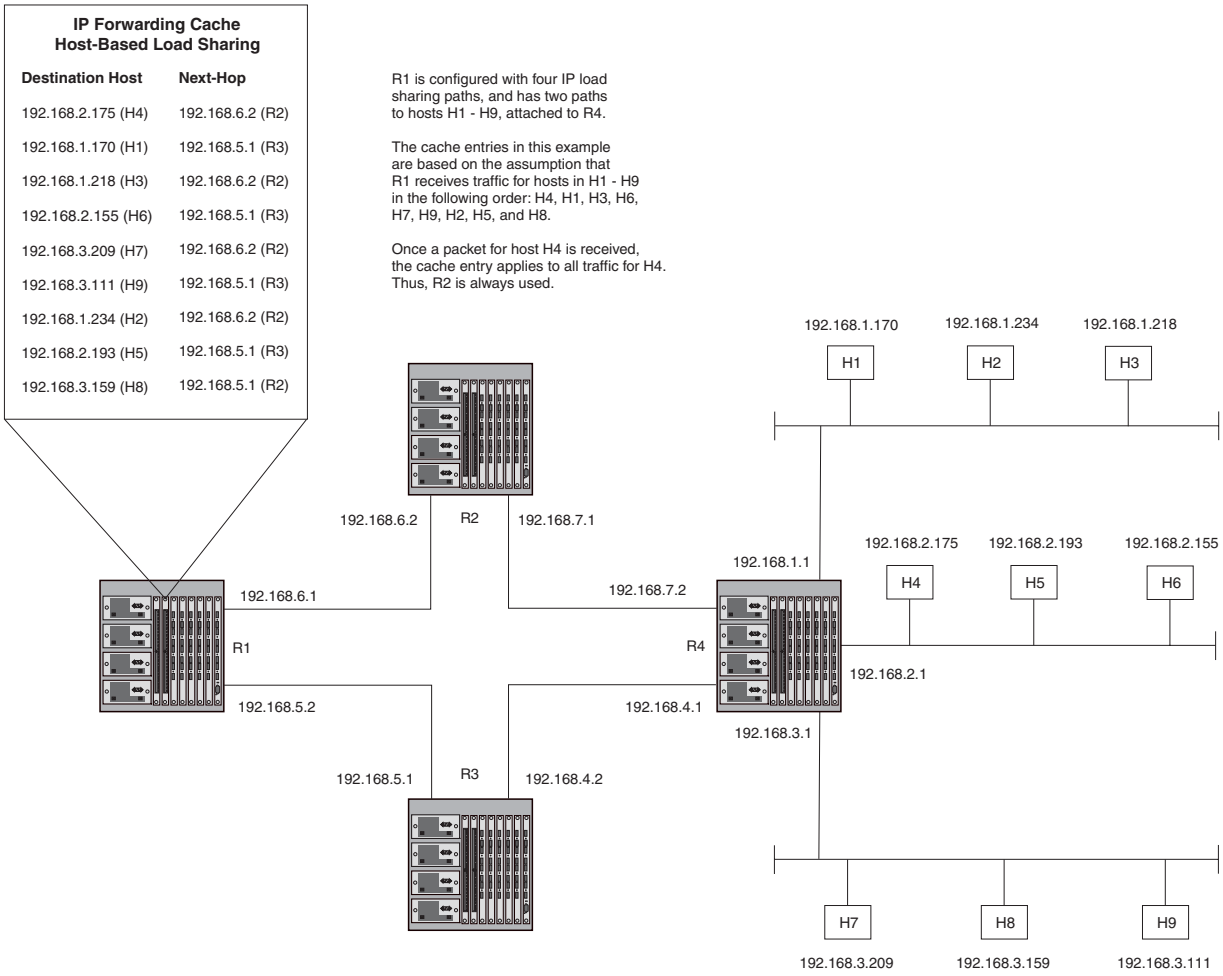


As shown in this example, when the Layer 3 Switch receives traffic for a destination and the IP route table has multiple equal-cost paths to that destination, the Layer 3 Switch selects the next equal-cost path (next-hop router) in the rotation and assigns that path to destination. The path rotation is determined by the order in which the IP route table receives the paths.

Since the configuration in this example contains two paths to hosts H1 – H9, the software alternates between the two paths when creating new load sharing cache entries for hosts H1 – H9. So long as the cache entry for a destination remains in the cache, the Layer 3 Switch always uses the same path for the traffic to the destination. In this example, the Layer 3 Switch always uses R2 as the next hop for forwarding traffic to H1.

Figure 12.9 shows another example of IP forwarding cache entries for the configuration shown in Figure 12.8. The network and load sharing configurations are the same, but the order in which R1 receives traffic for the host is different. The paths differ due to the order in which the Layer 3 Switch receives the traffic for the destination hosts.

Figure 12.9 Host-based IP load sharing – additional example



Network-Based IP Load Sharing

Network-based load sharing distributes traffic across multiple equal-cost paths based on the destination network. This method of load sharing optimizes system resources by aggregating the forwarding cache entries used for load sharing. Host-based load sharing contains a separate cache entry for each destination host, whereas network-based load sharing contains a single entry for each destination network.

The network-based load sharing method is available only on Chassis Layer 3 Switches and is the default.

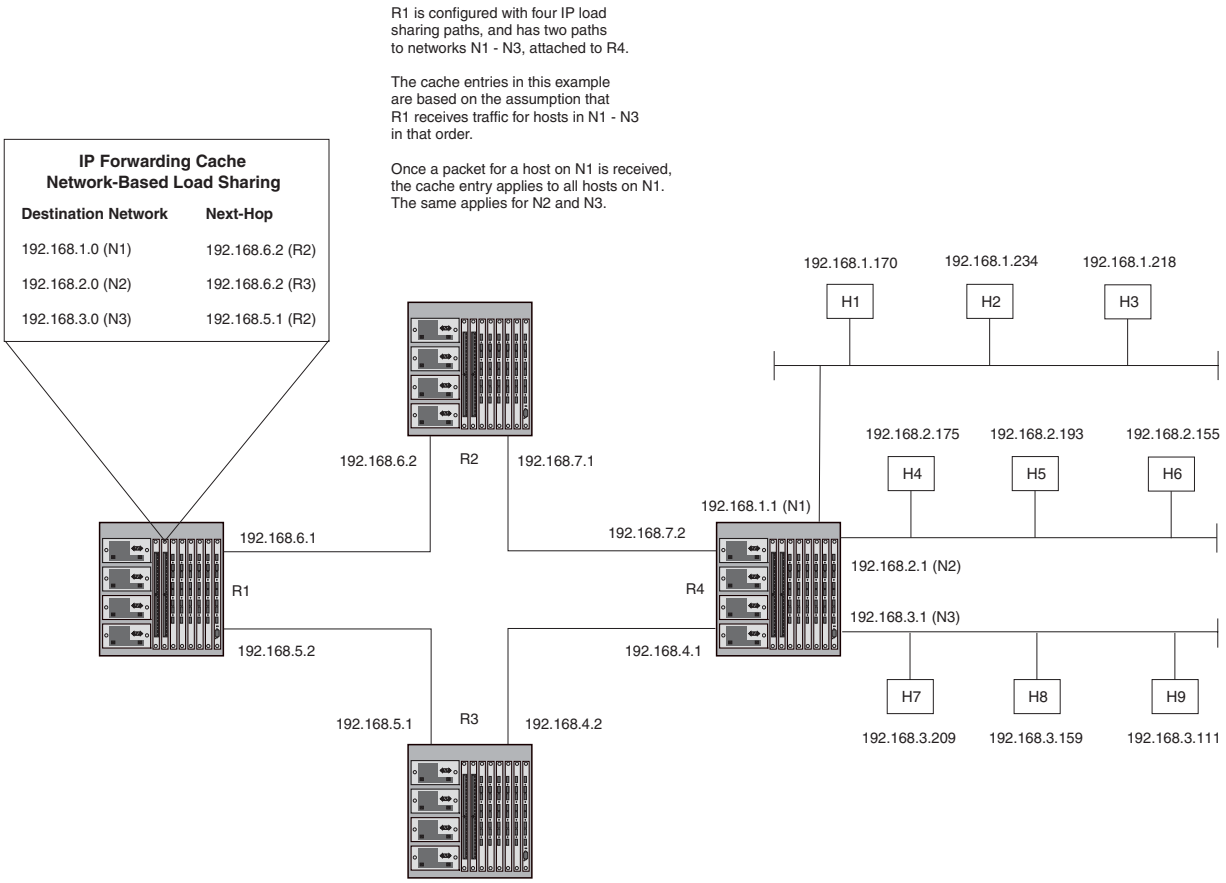
When the Layer 3 Switch receives traffic for a device on a destination network for which the IP route table has multiple equal-cost paths, the Layer 3 Switch checks the IP forwarding cache for a forwarding entry to the destination network:

- If the IP forwarding cache contains a forwarding entry for the destination network, the device uses the entry to forward the traffic.
- If the IP forwarding cache does not contain a forwarding entry for the destination network, the software selects

the next path in the rotation (the path after the one the software used for the previous load sharing selection). The software then creates an IP forwarding cache entry that associates the destination network address with the selected path. IP forwarding cache entries for network-based load sharing do not age out. Once the software creates a cache entry for a destination network, traffic for all hosts on the network uses the same path. The cache entries remain in effect until the state of one of the paths changes or the software is reloaded.

Figure 12.10 shows an example of IP load sharing cache entries for network-based IP load sharing. The network in this example is the same as the network in Figure 12.8 and Figure 12.9. Notice that the cache contains one entry for each destination network, instead of a separate entry for each destination host. Based on the cache entries, traffic for all hosts (H1, H2, and H3) on network N1 uses the path through R2.

Figure 12.10 Network-based IP load sharing – basic example



Notice that network-based load sharing does not use a simple round-robin method. The path rotation starts with path 2, then proceeds in ascending numerical order through the remaining paths and ends with path 1. In Figure 12.10, the first cache entry uses path 2 instead of path 1. The algorithm evenly distributes the load among the available paths, but starts with the second path instead of the first path.

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. See “Changing the Maximum Number of Load Sharing Paths” on page 12-79.

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.

The network-based IP load sharing mechanism selects a path based on the following calculation, which involves the maximum number of paths allowed on the Layer 3 Switch and the number of equal-cost paths available to the destination network.

M modulo P + 1 = S

where:

M = A number from 1 to the maximum number of load-sharing paths. This value increases by 1 until it reaches the maximum, then reverts to 1.

P = Number of equal-cost paths to destination network

S = Selected path

For reference, the following table lists the path that the network-based IP load sharing algorithm will select for each combination of maximum number of paths and number of actual paths to the destination network. The software orders the available paths based on when they enter the IP route table. The first path to enter the table is path 1, and so on.

The rows with maximum path value 4 list the path selections that occur using the default maximum number of load sharing paths, which is four.

Table 12.8: Path Selection for Network-Based IP Load Sharing

Number of Paths	Maximum Paths	Path Counter Value							
		1	**2**	**3**	**4**	**5**	**6**	**7**	**8**
2	2	2	1						
	3	2	1	2					
	4	2	1	2	1				
	5	2	1	2	1	2			
	6	2	1	2	1	2	1		
	7	2	1	2	1	2	1	2	
	8	2	1	2	1	2	1	2	1
3	2	2	3						
	3	2	3	1					
	4	2	3	1	2				
	5	2	3	1	2	3			
	6	2	3	1	2	3	1		
	7	2	3	1	2	3	1	2	
	8	2	3	1	2	3	1	2	3

Table 12.8: Path Selection for Network-Based IP Load Sharing (Continued)

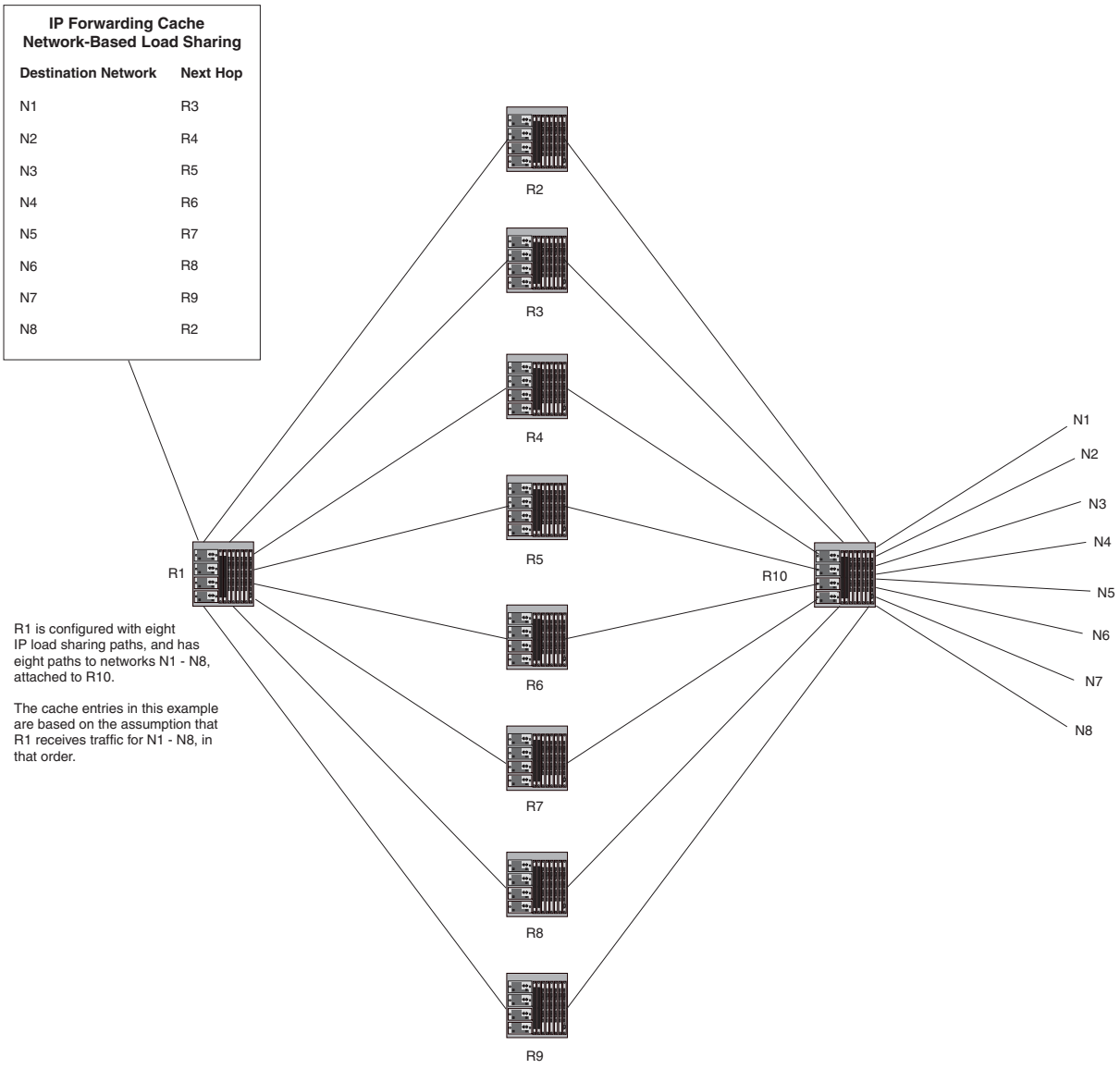
Number of Paths	Maximum Paths	Path Counter Value							
		1	**2**	**3**	**4**	**5**	**6**	**7**	**8**
4	2	2	3						
	3	2	3	4					
	4	2	3	4	1				
	5	2	3	4	1	2			
	6	2	3	4	1	2	3		
	7	2	3	4	1	2	3	4	
	8	2	3	4	1	2	3	4	1
5	2	2	3						
	3	2	3	4					
	4	2	3	4	5				
	5	2	3	4	5	1			
	6	2	3	4	5	1	2		
	7	2	3	4	5	1	2	3	
	8	2	3	4	5	1	2	3	4
6	2	2	3						
	3	2	3	4					
	4	2	3	4	5				
	5	2	3	4	5	6			
	6	2	3	4	5	6	1		
	7	2	3	4	5	6	1	2	
	8	2	3	4	5	6	1	2	3
7	2	2	3						
	3	2	3	4					
	4	2	3	4	5				
	5	2	3	4	5	6			
	6	2	3	4	5	6	7		
	7	2	3	4	5	6	7	1	
	8	2	3	4	5	6	7	1	2

Table 12.8: Path Selection for Network-Based IP Load Sharing (Continued)

Number of Paths	Maximum Paths	Path Counter Value							
		1	**2**	**3**	**4**	**5**	**6**	**7**	**8**
8	2	2	3						
	3	2	3	4					
	4	2	3	4	5				
	5	2	3	4	5	6			
	6	2	3	4	5	6	7		
	7	2	3	4	5	6	7	8	
	8	2	3	4	5	6	7	8	1

As shown in Table 12.8, the results of the network-based IP load sharing algorithm provide evenly-distributed load sharing. Figure 12.11 shows a network where a Layer 3 Switch has eight equal-cost paths to destination networks N1 – N8. The Layer 3 Switch (R1) has been enabled to support up to eight IP load sharing paths.

Figure 12.11 Network-based IP load sharing – example with eight equal-cost paths and eight destination networks



As shown in this example, the algorithm for network-based IP load-sharing does not select the paths beginning with the first path, but the algorithm nonetheless results in an evenly distributed selection of paths.

Disabling or Re-Enabling Load Sharing

If you do not use IP load sharing and you want to disable the feature, use either of the following methods.

USING THE CLI

To disable IP load sharing, enter the following commands:

```
BigIron(config)# no ip load-sharing
```

Syntax: [no] ip load-sharing

USING THE WEB MANAGEMENT INTERFACE

1. Log on to the Layer 3 Switch using a valid user name and password for read-write access. The System configuration panel is displayed.

2. Click on the plus sign next to Configure in the tree view to expand the list of configuration options.
3. Click on the plus sign next to IP in the tree view to expand the list of IP option links.
4. Click on the General link to display the IP configuration panel.
5. Click the Disable radio button next to Load Sharing.
6. Click the Apply button to save the change to the device's running-config file.
7. 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.

Changing the Load Sharing Method on Chassis Layer 3 Switches

Chassis Layer 3 Switches can perform IP load sharing based on destination host address or destination network address. The default for all Chassis Layer 3 Switches is network-based IP load sharing. If you want to enable a Chassis Layer 3 Switch to perform host-based IP load sharing instead, use either of the following methods.

NOTE: Stackable Layer 3 Switches support host-based IP load sharing only.

NOTE: Regardless of the method of load sharing that is enabled on a Chassis Layer 3 Switch, the Layer 3 Switch always load shares paths for default routes and the network default route based on destination host address.

NOTE: The VM1 uses hash-based load balancing instead of host-based or network-based load balancing. A hash value is calculated based on the source and destination IP addresses. Each of the paths to a given destination is associated with one of the possible hash values, and the traffic flow is assigned to a path based on its calculated hash value.

USING THE CLI

To enable host-based IP load sharing, enter the following command:

```
BigIron(config)# ip load-sharing by-host
```

This command enables host-based IP load sharing on the device. The command also disables network-based IP load-sharing at the same time.

Syntax: [no] ip load-sharing by-host

To disable host-based IP load sharing and re-enable network-based IP load sharing, enter the following command:

```
BigIron(config)# no ip load-sharing by-host
```

NOTE: The VM1 uses hash-based load balancing regardless of the type of IP load sharing enabled (**by-host** or **by-network**).

USING THE WEB MANAGEMENT INTERFACE

You cannot configure this option using the Web management interface.

Enabling Host-Based Load-Sharing for a Specific Destination Network

Chassis Layer 3 Switches can perform IP load sharing on a network basis or an individual host basis. The default on these devices is network-based load sharing. You can take advantage of the forwarding-cache optimization provided by network-based load sharing while using the more granular host-based load sharing for specific destination networks.

Use this feature when you want to use network-based load sharing by default but also want to use host-based load sharing for specific destination networks.

NOTE: This feature applies only to Chassis Layer 3 Switches. Stackable Layer 3 Switches perform host-based load sharing for all destinations and cannot be configured for network-based load sharing. Use this feature only when network-based load sharing is enabled.

When you configure host-based load sharing for a specific destination network, the Layer 3 Switch distributes traffic to hosts on the network evenly across the available paths. For other networks, the Layer 3 Switch uses a single path for all traffic to hosts on a given network.

NOTE: The host-based load sharing for the destination takes effect only if the IP route table contains an entry that exactly matches the destination network you specify. For example, if you configure host-based load sharing for destination network 207.95.7.0/24, the IP route table must contain a route entry for that network. In fact, for load sharing to occur, the IP route table needs to contain multiple equal-cost paths to the network.

To enable host-based load sharing for a specific destination network, use the following CLI method.

USING THE CLI

To enable host-based load sharing for a specific destination network, enter a command such as the following at the global CONFIG level of the CLI:

```
BigIron(config)# ip load-sharing route-by-host 207.95.7.0/24
```

This command configures the Layer 3 Switch to use host-based load sharing for traffic to destinations on the 207.95.7.0/24 network. The Layer 3 Switch uses network-based load sharing for traffic to other destination networks.

Syntax: [no] ip load-sharing route-by-host <ip-addr> <ip-mask>

or

Syntax: [no] ip load-sharing route-by-host <ip-addr>/<mask-bits>

USING THE WEB MANAGEMENT INTERFACE

You cannot configure this option using the Web management interface.

Disabling Host-Based Load-Sharing

You can disable host-based load sharing for specific destination networks or for all networks. When you disable host-based load sharing for a destination network (or for all destination networks), the software removes the host-based forwarding cache entries for the destination network(s) and uses network-based forwarding entries instead.

NOTE: This method applies only to networks for which you have explicitly enabled host-based load sharing. If you have enabled host-based load sharing globally but want to change to network-based load sharing, enter the no ip load-sharing by-host command at the global CONFIG level of the CLI.

Use either of the following methods to disable host-based load sharing for destination networks for which you have configured the feature.

USING THE CLI

To disable host-based load sharing for all the destination networks for which you have explicitly enabled the host-based load sharing, enter the following command at the global CONFIG level of the CLI:

```
BigIron(config)# no ip load-sharing route-by-host
```

To disable host-based load sharing for a specific destination network, enter a command such as the following:

```
BigIron(config)# no ip load-sharing route-by-host 207.95.7.0/24
```

This command removes the host-based load sharing for the 208.95.7.0/24 network, but leaves the other host-based load sharing configurations intact.

USING THE WEB MANAGEMENT INTERFACE

You cannot configure this option using the Web management interface.

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 – 8.

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 paths, use either of the following methods.

USING THE CLI

To change the number of IP load sharing paths, enter a command such as the following:

```
BigIron(config)# ip load-sharing 8
```

Syntax: [no] ip load-sharing [<num>]

The <num> parameter specifies the number of paths and can be from 2 – 8.

USING THE WEB MANAGEMENT INTERFACE

1. Log on to the Layer 3 Switch using a valid user name and password for read-write access. The System configuration panel is displayed.
2. Click on the plus sign next to Configure in the tree view to expand the list of configuration options.
3. Click on the plus sign next to IP in the tree view to expand the list of IP option links.
4. Click on the General link to display the IP configuration panel.
5. Edit the value in the # of Paths field. You can enter a number from 2 – 8.
6. Click the Apply button to save the change to the device's running-config file.
7. 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.

IP Load Sharing for RIPv2 Routes

In release 08.0.00, IP load sharing is supported for RIPv2 routes. IP load sharing is also called "Equal-Cost Multi-Path (ECMP)" load sharing or just "ECMP".

In previous releases, the Foundry device did not keep equal-cost routes from different next hops. Only the last route received for a network was kept. Starting with this release, the device stores multiple equal-cost RIPv2 routes to the same destination, and shares the traffic load among the routes.

By default, IP load sharing for RIPv2 routes is disabled. To enable it, enter the following commands:

```
BigIron(config)# router rip
BigIron(config-rip-router)# ecmp-enable
```

Syntax: [no] ecmp-enable

Optimizing the IP Forwarding Cache

NOTE: This section applies only to Chassis Layer 3 Switches.

Chassis Layer 3 Switches use ***Content Addressable Memory (CAM)*** as a fast lookup cache to optimize IP forwarding. The CAM contains an IP route's destination and the IP address of the next-hop gateway, as well as

pointers to packet information in various system buffers. When the Layer 3 Switch is ready to forward a packet to its destination, the Layer 3 Switch checks the CAM for a forwarding entry for the packet.

- If the CAM contains an entry, the Layer 3 Switch uses the entry to forward the packet.
- If the CAM does not contain an entry, the Layer 3 Switch searches the IP route table for a route to the packet's destination, then programs an entry into the CAM for the destination and its next-hop gateway. The Layer 3 Switch uses the CAM entry to forward the next packet to this destination.

By default, the CAM is optimized for environments with a lot of routes to different destination networks. Each CAM entry provides fast-path information for a different destination subnet.

You can configure the following cache and CAM optimization options:

- **ip hi-perf** – Enables the cache to contain more unique host route entries for unicast traffic.

  NOTE: This feature is enabled by default in software release 07.5.01 and later.

- The following features optimize the CAM for devices that have very large IP route tables (100,000 or more), where most of those routes use the same next hops as the default route.
  - **ip net-aggregate** – Divides the IP address space into 4096 (2^{12}) aggregate entries and applies a 12-bit prefix to each aggregate entry.
  - **ip net-aggregate premium** – Divides the IP address space into 8192 (2^{13}) aggregate entries and applies a 13-bit prefix to each aggregate entry. This feature is supported in software releases 07.8.00 and later. See "Increased CAM Network Aggregation in Release 07.8.00" on page 12-81.
  - **ip net-aggregate supreme** – Divides the IP address space into 16,384 (2^{14}) aggregate entries and applies a 14-bit prefix to each aggregate entry. This feature is supported in software releases 07.8.00 and later. See "Increased CAM Network Aggregation in Release 07.8.00" on page 12-81.
- **ip dr-aggregate** – Optimizes the CAM for devices that have few explicit routes (about 30 or fewer) and use the default route for most of the traffic.

Regardless of whether one of the CAM optimization options described above is enabled, the Layer 3 Switch uses the IP cache to store forwarding information, then uses the forwarding information in the IP cache to program the CAM. The IP cache can contain host route entries, network route entries, and aggregate entries (aggregate routes of varying prefix lengths or fixed-size portions of the default route). However, regardless of the CAM optimization options, the **show ip cache** command displays only the host route entries.

Disabling Unicast High-Performance Mode

By default, the unicast high-performance mode is enabled. This mode increases the device's capacity for unicast entries. To disable or re-enable the feature, use the following CLI method.

NOTE: To place a change to the high-performance mode into effect, you must reload the software after saving the change to the startup-config file.

NOTE: The feature is disabled by default in software releases earlier than 07.5.01.

USING THE CLI

To disable the high-performance mode, enter the following commands:

```
BigIron(config)# no ip high-perf
BigIron(config)# write memory
BigIron(config)# end
BigIron# reload
```

To enable the high-performance mode, enter the following commands:

```
BigIron(config)# ip high-perf
BigIron(config)# write memory
```



```
BigIron(config)# end
BigIron# reload
```

Syntax: [no] ip high-perf

CAM Optimization Options

By default, the device programs a separate route entry for each destination network. For example, if the device forwards traffic to 10.0.0.0/24, 10.0.10.0/24, and 10.10.0.20/24, the device programs a separate CAM entry for each of the three destination networks.

In many cases, the default behavior provides optimal routing performance. However, you may want to enable a CAM optimization option if either of the following conditions is true:

- The device has a very large IP route table (100,000 routes or more). For example, this can occur if the device is a BGP4 router and contains a full set of BGP4 routes. If most of the BGP4 routes actually go to the same set of next hops as the default route, enable the CAM network aggregation feature.
- The device has relatively few explicit routes in the IP route table and uses the default route for all other traffic. In this case, enable the CAM default route aggregation feature.

With CAM network aggregation enabled, the device forward IP traffic as follows:

The device checks the CAM for an entry with the traffic's destination.

- If the CAM contains an entry, the device uses the entry.
- If the CAM does not contain an entry, the device checks to see whether all explicit routes in the IP route table that are in the same aggregate (for example, /12) as the needed route (all routes that overlap with the /12 aggregate), have the same set of next hops as the default route.
    - If all explicit routes in the IP route table that are within the same aggregate use the same next hops as the default route, the device programs a single CAM entry that aggregates the route information for all routes within the aggregate. The device uses this single CAM entry to forward traffic to any destination within the aggregate.
    - If one or more explicit routes within the same aggregate uses a next hop that is not also used by the default route, the device does not program an aggregate entry into the CAM but instead programs a separate route entry for the individual destination network.

After programming a CAM entry for the traffic's destination, the device uses the entry to forward further traffic to the same destination. If the device was able to program an aggregate entry, the device uses the entry for traffic to any destination within the aggregate.

NOTE: CAM network aggregation requires a default route in the IP route table.

Increased CAM Network Aggregation in Release 07.8.00

NOTE: This feature applies to JetCore and 10-Gigabit Ethernet devices.

- In releases prior to 07.8.00, when you enable CAM network aggregation (**ip net-aggregate** command), the maximum number of aggregate entries is 4096 and each aggregate entry has a 12-bit prefix (/12).
- Starting in release 07.8.00, on JetCore and 10-Gigabit Ethernet devices, you can increase the number of aggregate entries on a Foundry device from a maximum of 4096 (the default), to a maximum of 8192 or 16,384 entries, depending on the Layer 3 CAM space available on the device. The device automatically adjusts the prefix for each aggregate entry, according to the maximum number of aggregate entries allowed on the device, as shown in Table 12.9.

The following table shows the CAM network aggregation support in releases 07.8.00 and above on JetCore and 10-Gigabit Ethernet devices.

Table 12.9: CAM Network Aggregation Support

Description	Maximum Number of Aggregate Entries	Prefix
Standard CAM optimization (default)	4096	/12
Premium CAM optimization	8192	/13
Supreme CAM optimization	16,384	/14

One advantage to an increased number of aggregate entries is better load balancing for outgoing traffic, since most of the routes use the default route's set of next hops rather than a single next hop destination. In addition, since JetCore and 10-Gigabit Ethernet modules have extensive Layer 3 CAM space, the software can allocate more aggregate entries without compromising the space needed for entries that cannot be aggregated.

Configuration Notes

- Premium and Supreme CAM optimization features are supported in software releases 07.8.00 and later.
- Premium and Supreme CAM optimization features are supported on devices with JetCore and 10-Gigabit Ethernet only.
- Before configuring Premium and Supreme CAM optimization features, you must check the Layer 3 CAM on all interface modules to ensure there is enough space to hold as many routes (aggregate as well as non-aggregate routes) as possible. To check the Layer 3 CAM, use the **show cam-partition** command. To increase the Layer 3 CAM partition, use the **cam-partition** command.
- If you have a mixture of modules in a Foundry chassis device, the software uses the lowest common denominator to determine the maximum number of supported CAM network aggregate entries. For example, on a FastIron device, the software will use 4K CAM since this is the lowest common denominator on the device.

Maximum Number of Aggregate Entries per Module

The maximum number of CAM network aggregate entries supported on your device depends on the availability of, and the constraints on, the Layer 3 CAM. When you enable CAM network aggregation, the device will increase the number of CAM network aggregate entries up to a maximum of 50% of the IP route (Layer 3) CAM space, excluding IP supernet and IPX networks.

For example, as shown in Table 12.10, in order to increase the number of aggregate entries to 8000 on a BigIron device, the device must have a minimum of 16,000 CAM entries available in Layer 3 CAM, excluding the IPX partition and the IP supernet areas. Similarly, in order to increase the number of CAM network aggregate entries to 16,000 on a BigIron JetCore-based device, the IP Layer 3 CAM must be capable of holding at least 32,000 aggregate entries.

Note that CAM space on IronCore modules is limited and is also shared between two or more DMAs. Adversely, CAM space on JetCore modules is larger because these devices use Ternary CAM (TCAM), which is capable of being partitioned. The CAM space on 10-Gigabit Ethernet modules is similar to that on JetCore and the overall space is even larger.

Table 12.10 shows the maximum number of supported aggregate entries per Foundry module.

Table 12.10: Maximum Number of Aggregate Entries per Module

Module	Device	Minimum Number of CAM Entries Available in Layer 3 CAM	Maximum Supported CAM Network Aggregate Entries[1]
JetCore	BigIron	32K	16K
	FastIron	16K	8K

Table 12.10: Maximum Number of Aggregate Entries per Module

Module	Device	Minimum Number of CAM Entries Available in Layer 3 CAM	Maximum Supported CAM Network Aggregate Entries[1]
IronCore	BigIron	16K	8K
	FastIron	8K	4K
10 Gigabit Ethernet	BigIron and FastIron	64K	16K

1.The numbers shown are per DMA.

Enabling Standard Optimization for CAM Network Aggregation

When you enable standard optimization of CAM network aggregation, the software divides the IP address space into 4096 (2^{12}) aggregate entries and applies a 12-bit prefix to each aggregate entry.

To enable standard CAM network aggregation, enter the following command at the global CONFIG level of the CLI:

```
BigIron(config)# ip net-aggregate
```

Syntax: [no] ip net-aggregate [<secs>]

Enabling Premium Optimization for CAM Network Aggregation

When you enable premium optimization of CAM network aggregation, the software divides the IP address space into 8192 (2^{13}) aggregate entries and applies a 13-bit prefix to each aggregate entry.

NOTE: This feature is supported in releases 07.8.00 and later.

To enable premium optimization for CAM network aggregation, enter the following command at the global CONFIG level of the CLI:

```
BigIron(config)# ip net-aggregate premium
```

Syntax: [no] ip net-aggregate premium

Enabling Supreme Optimization for CAM Network Aggregation

When you enable supreme optimization of CAM network aggregation, the software divides the IP address space into 16,384 (2^{14}) aggregate entries and applies a 14-bit prefix to each aggregate entry.

NOTE: This feature is supported in releases 07.8.00 and later.

To enable supreme CAM network aggregation, enter the following command at the global CONFIG level of the CLI:

```
BigIron(config)# ip net-aggregate supreme
```

Syntax: [no] ip net-aggregate supreme

Displaying CAM Network Aggregation Entries

To display the CAM network aggregation entries, enter the following command at any level of the CLI:

```
BigIron(config)# show ip net-aggregate
Total prefixes: 4096, CAM Ineligible: 31, Setups: 14, Updates 7477
Start index: 1
0.0.0.0/13        Gateway: 101.77.7.101
CAM Entry Flag: 00000703H
CIDX0: 5553  CIDX8: 4127
0.16.0.0/13        Gateway: 101.78.7.101
CAM Entry Flag: 00000003H
CIDX0: 5552
0.32.0.0/13        Gateway: 101.79.7.101
CAM Entry Flag: 00000003H
CIDX0: 5551
0.48.0.0/13        Gateway: 101.76.7.101
CAM Entry Flag: 00000003H
CIDX0: 5550
0.64.0.0/13        Gateway: 101.77.7.101
CAM Entry Flag: 00000003H
CIDX0: 5549
0.80.0.0/13        Gateway: 101.78.7.101
CAM Entry Flag: 00000003H
CIDX0: 5548
0.96.0.0/13        Gateway: 101.79.7.101
CAM Entry Flag: 00000003H
CIDX0: 5547
0.112.0.0/13        Gateway: 101.76.7.101
CAM Entry Flag: 00000003H
CIDX0: 5546
--More--, next page: Space, next line: Return key, quit: Control-c
```

This example shows that premium optimization for CAM network aggregation is enabled. The default-route optimization feature divides the default route into individual networks with 13-bit prefixes. The first entry is network 0.0.0.0/13, the second entry is network 0.16.0.0/13, and so on.

Syntax: show ip net-aggregate [<starting-entry-num> | <ip-addr> | not-eligible]

The <starting-entry-num> specifies the entry number you want the command's output to start with. By default, the display begins with the first entry.

The <ip-addr> parameter specifies the IP address of a destination. The CAM entry that contains the specified address is displayed.

The **not-eligible** parameter displays only the entries that are ineligible for use because they contain a destination network that the Layer 3 Switch uses a route other than the default route to reach.

The **show ip net-aggregate** command displays the following information.

Table 12.11: CLI Display of CAM

This Field...	Displays...
Total prefixes	The total number of entries in the CAM.

Table 12.11: CLI Display of CAM (Continued)

This Field...	Displays...
CAM Ineligible	The number of entries that cannot be used for fast-path forwarding because the IP route table contains a route whose destination network is contained in the entry's aggregate network, but does not use the default route.
Setups	The number of times the entire CAM has been reprogrammed during the current power cycle. Generally, this occurs when the default route changes.
Updates	The number of individual entries that have been updated due during the current power cycle to a route change.
Start index	The entry number of the first entry in the display. If you specify a starting entry number when you enter the **show ip net-aggregate** command, then this field shows that number. Otherwise, the starting number is 1.
Destination address	An aggregate network address. If a route's destination is contained in this aggregate address, then this CAM entry is applicable to the destination. **Note**: When CAM network aggregation is enabled, the entry is actually used only if the Layer 3 Switch uses the default route to reach the destination.
Gateway	The IP address of the next-hop gateway reached through the default route. **Note**: The default route can have more than one next-hop gateway address. When this is the case, the Layer 3 Switch load balances traffic across the gateways using the IP load sharing settings in effect in the software. For information, see the "Configuring IP Load Sharing" on page 12-66.
CAM Entry Flag	A value used by Foundry Technical Support for troubleshooting.
CIDX*n*	A value used by Foundry Technical Support for troubleshooting.

CAM Default Route Aggregation

The CAM default route aggregation feature optimizes CAM use in environments that have relatively few explicit routes in the route table and use a default route heavily.

Without CAM default route aggregation, the device programs a CAM entry for each destination that uses an explicit route in the route table and also programs a separate CAM entry for each destination that uses the default route. For example, suppose the IP route table contains two explicit routes, 20.0.0.x and 30.0.0.x and uses the default route for all other destinations. When the device needs to forward traffic to 20.0.0.x, the device uses the existing CAM entry for the destination. If this is the first time the device is forwarding traffic to the destination and the CAM entry therefore hasn't been programmed yet, the device programs the entry for 20.0.0.x.

The same process occurs for traffic destined to a network that doesn't have an explicit route in the IP route table. When the device needs to forward traffic to a destination that requires the default route, the device creates a CAM

entry for the destination network. For example, if the device needs to forward traffic to 40.40.40.x and 40.41.41.x, the device creates two CAM entries, one for 40.40.40.x and another for 40.41.41.x.

When the device needs to forward traffic on the default route, the device attempts to build an aggregate route that does not conflict with an explicit route in the IP route table. (A conflict occurs if an explicit host route in the table overlaps with the aggregate.)

For example, with CAM default route aggregation enabled, the device creates a single CAM entry, 40.0.0.0/8, for 40.40.40.x and 40.41.41.x. In fact, traffic for any network that overlaps with 40.0.0.0/8 uses the same CAM entry.

The device begins with a /8 aggregate.

- If there are no conflicts with explicit routes, the device programs the /8 aggregate into the CAM.
- If there is a conflict, the device tries a /12 aggregate, and so on in increments of 4 (/16, /20, /24, and so on) until a non-conflicting entry can be programmed into the CAM.

NOTE: CAM default route aggregation requires a default route in the IP route table.

Enabling CAM Default Route Aggregation

To enable CAM default route optimization, enter the following command at the global CONFIG level of the CLI:

```
BigIron(config)# ip dr-aggregate
```

Syntax: [no] ip dr-aggregate

To disable the default-route aggregation mode, enter the following command:

```
BigIron(config)# no ip dr-aggregate
```

Displaying the CAM Default Route Aggregation CAM Entries

To display the CAM default route aggregation entries, enter the following command at any level of the CLI:

```
BigIron(config)# show ip dr-aggregate
```

Syntax: show ip dr-aggregate [<ip-addr>]

If you specify an IP address, only the entries for that destination are displayed.

Here is an example of the information displayed by this command.

```
BigIron(config)# show ip dr-aggregate
Total number of cache entries: 2
Start index: 1  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     22.22.22.22    /8  207.95.6.60     0044.052e.4302  DF    1/1   1     0
2     207.96.7.7     /12 207.95.6.60     0044.052e.4302  DF    1/1   1     0
```

This example shows two entries. The prefix associated with each entry is displayed. Notice that the prefix lengths in this example are different for each entry. The software selects a prefix length long enough to make the default network route entry unambiguous, so that is does not conflict with other cache entries.

To display the entry for a specific destination, enter the destination address, as shown in the following example.

```
BigIron(config)# show ip dr-aggregate 207.96.7.7
Total number of cache entries: 2
Start index: 1  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     207.96.7.7     /12 207.95.6.60     0044.052e.4302  DF    1/1   1     0
```


This example shows the second entry from the previous example, but the entry row number is 1. The row number identifies the row number in the displayed output. In addition, notice that the Total number of cache entries field shows 2, as in the previous example. The number in this field indicates the total number of default route aggregation entries in the forwarding cache.

Clearing the Forwarding Cache Entries for Default Routes

To clear the entries, enter the following command from the Privileged EXEC level of the CLI:

```
BigIron# clear ip dr-aggregate
```

Syntax: clear ip dr-aggregate

NOTE: This command does not affect other types of forwarding cache entries.

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 enable IRDP globally, use either of the following methods.

USING THE CLI

To globally enable IRDP, enter the following command:

```
BigIron(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.

USING THE WEB MANAGEMENT INTERFACE

1. Log on to the Layer 3 Switch using a valid user name and password for read-write access. The System configuration panel is displayed.
2. Click on the plus sign next to Configure in the tree view to display the list of configuration options.
3. Click on the plus sign next to IP to display the list of IP configuration options.
4. Select the General link to display the IP configuration panel.
5. Select Enable next to IRDP.
6. Click the Apply button to save the change to the device's running-config.
7. 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.

Enabling IRDP on an Individual Port

To enable IRDP on an individual port and configure IRDP parameters, use either of the following methods.

USING THE CLI

To enable IRDP on an individual interface and change IRDP parameters, enter commands such as the following:

```
BigIron(config)# interface ethernet 1/3
BigIron(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.

USING THE WEB MANAGEMENT INTERFACE

You cannot configure these options using the Web management interface.

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 Forwarding Parameters" on page 12-95.

Disabling RARP

RARP is enabled by default. If you want to disable the feature, you can do so using either of the following methods.

USING THE CLI

To disable RARP, enter the following command at the global CONFIG level:

```
BigIron(config)# no ip rarp
```

Syntax: [no] ip rarp

To re-enable RARP, enter the following command:

```
BigIron(config)# ip rarp
```

USING THE WEB MANAGEMENT INTERFACE

1. Log on to the Layer 3 Switch using a valid user name and password for read-write access. The System configuration panel is displayed.
2. Click on the plus sign next to Configure in the tree view to expand the list of configuration options.
3. Click on the plus sign next to IP in the tree view to expand the list of IP option links.
4. Click on the General link to display the IP configuration panel.
5. Select the Disable or Enable radio button next to RARP.
6. Click the Apply button to save the change to the device's running-config file.
7. 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.

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 configure static RARP entries, use the following methods.

USING THE CLI

To assign a static IP RARP entry for static routes on a Foundry router, enter a command such as the following:

```
BigIron(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>

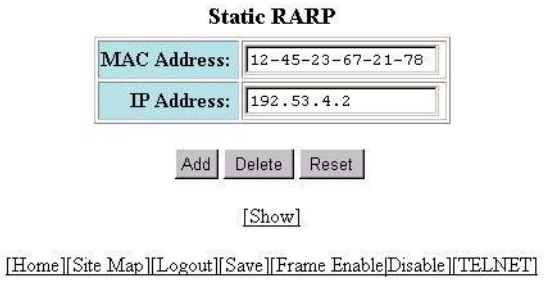
The <number> parameter identifies the RARP entry number. You can specify an unused number from 1 to the maximum number of RARP entries supported on the device. To determine the maximum number of entries supported on the device, see the "Displaying and Modifying System Parameter Default Settings" section of the "Configuring Basic Features" chapter of the *Foundry Switch and Router Installation and Basic Configuration Guide* or the *Foundry BigIron Mg8 Switch Installation and Basic Configuration Guide*.

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.

USING THE WEB MANAGEMENT INTERFACE

1. Log on to the Layer 3 Switch using a valid user name and password for read-write access. The System configuration panel is displayed.
2. Click on the plus sign next to Configure in the tree view to expand the list of configuration options.
3. Click on the plus sign next to IP in the tree view to expand the list of IP option links.
4. Click on the General link to display the IP configuration panel.
5. Click the Static RARP link.
   - If the device does not have any static RARP entries, the Static RARP configuration panel is displayed, as shown in the following example.
   - If a static RARP entry is already configured and you are adding a new entry, click on the Add Static RARP link to display the Static RARP configuration panel, as shown in the following example.
   - If you are modifying an existing static RARP entry, click on the Modify button to the right of the row describing the entry to display the Static RARP configuration panel, as shown in the following example.



6. Enter the MAC address.
7. Enter the IP address.
8. Click the Add button to save the change to the device's running-config file.
9. 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.

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 "Displaying and Modifying System Parameter Default Settings" section of the "Configuring Basic Features" chapter of the *Foundry Switch and Router Installation and Basic Configuration Guide* or the *Foundry BigIron Mg8 Switch Installation and Basic Configuration Guide*.

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: As shown above, 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 Forwarding Parameters" on page 12-95.

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 either of the following methods. You also can disable forwarding for an application using these methods.

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 12-96.

USING THE CLI

To enable the forwarding of SNMP trap broadcasts, enter the following command:

```
BigIron(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:

```
BigIron(config)# no ip forward-protocol udp snmp
```

This command disables forwarding of SNMP requests to the helper addresses configured on Layer 3 Switch interfaces.

USING THE WEB MANAGEMENT INTERFACE

1. Log on to the Layer 3 Switch using a valid user name and password for read-write access. The System configuration panel is displayed.
2. Click on the plus sign next to Configure in the tree view to expand the list of configuration options.
3. Click on the plus sign next to IP in the tree view to expand the list of IP option links.
4. Click on the General link to display the IP configuration panel.
5. Select the Disable or Enable radio button next to Broadcast Forward.
6. Click the Apply button to save the change to the device's running-config file.
7. 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.

NOTE: To define the ports to be forwarded, select the UDP Helper link from the IP configuration sheet.

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, use either of the following methods.

USING THE CLI

To configure a helper address on interface 2 on chassis module 1, enter the following commands:

```
BigIron(config)# interface e 1/2
BigIron(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>

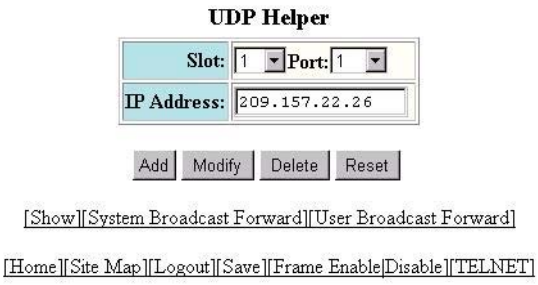
The <num> parameter specifies the helper address number and can be from 1 – 16.

The <ip-addr> command specifies the server's IP address or the subnet directed broadcast address of the IP subnet the server is in.

USING THE WEB MANAGEMENT INTERFACE

To configure a helper address on an interface:

1. Log on to the Layer 3 Switch using a valid user name and password for read-write access. The System configuration panel is displayed.
2. Click on the plus sign next to Configure in the tree view to expand the list of configuration options.
3. Click on the plus sign next to RIP in the tree view to expand the list of RIP option links.
4. Click on the UDP Helper link.
   - If the device does not have any UDP helper assignments, the UDP Helper configuration panel is displayed, as shown in the following example.
   - If a UDP helper assignment is already configured and you are adding a new one, click on the Add UDP Helper link to display the UDP Helper configuration panel, as shown in the following example.
   - If you are modifying an existing UDP helper assignment, click on the Modify button to the right of the row describing the assignment to display the UDP Helper configuration panel, as shown in the following example.



5. Select the port (and slot if applicable) on behalf of which the UDP helper packets will be forwarded.
6. Enter the IP address of the remote server for which the router will be relaying the packets.
7. Click the Add button to save the change to the device's running-config file.
8. To configure settings for another port, select the port (and slot, if applicable) and go to step 6.
9. 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.

To select an application to be forwarded to the server by the Layer 3 Switch:

1. Log on to the device using a valid user name and password for read-write access. The System configuration panel is displayed.

2. Click on the plus sign next to Configure in the tree view to expand the list of configuration options.
3. Click on the plus sign next to RIP in the tree view to expand the list of RIP option links.
4. Click on the UDP Helper link.
5. Click on the Modify button to the right of the row describing the UDP helper assignment to display the UDP Helper configuration panel.
6. Click on the System Broadcast Forward or User Broadcast Forward link.
   - The System Broadcast Forward link displays a panel that lets you select a well-known UDP port.
   - The User Broadcast Forward link displays a panel that lets you enter any port number.
7. Select the port or enter a port number from 1 – 65535.
8. Click the Add button to save the change to the device's running-config file.
9. 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.

Configuring Egress Priority Merging

By default, egress priority merging is enabled by default. The Foundry internal priority (calculated on ingress) is merged with the incoming VLAN tag priority at the egress on every port. This means that the highest value of these two priorities is used to set the outgoing packet priority.

The problem with this approach is that if an inbound ACL downgrades the priority (it does this by setting the internal priority to a lower value) it will not be honored for tagged packets, since the egress priority merge will override what the ACL attempted to do.

On the BigIron MG8 and NetIron 40G running release 02.2.01 and later, the egress priority merge is disabled by default. It must be enabled on each port, as desired. You can enable or disable egress priority merging on a port by entering the following command:

```
BigIron MG8(config)#interface ethernet 1/2
BigIron MG8(config-if-e10000-1/2)#merge-egress-priority
```

This will enable egress priority merging on the interface.

Syntax: [no] merge-priority

Configuring BootP/DHCP Forwarding 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 Forwarding 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 12-93.

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.

USING THE CLI

To change the IP address used for stamping BootP/DHCP requests received on interface 1/1, enter commands such as the following:

```
BigIron(config)# int e 1/1
BigIron(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>

USING THE WEB MANAGEMENT INTERFACE

You cannot change the IP address used for stamping BootP/DHCP requests using the Web management interface.

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.

USING THE CLI

To modify the maximum number of BootP/DHCP hops, enter the following command:

```
BigIron(config)# bootp-relay-max-hops 10
```

This command allows the Layer 3 Switch to forward BootP/DHCP requests that have passed through up to ten previous hops before reaching the Layer 3 Switch.

Syntax: bootp-relay-max-hops <1-15>

USING THE WEB MANAGEMENT INTERFACE

To modify the maximum number of hops supported:

1. Log on to the Layer 3 Switch using a valid user name and password for read-write access. The System configuration panel is displayed.
2. Click on the plus sign next to Configure in the tree view to display the list of configuration options.
3. Click on the plus sign next to IP to display the list of IP configuration options.
4. Select the General link to display the IP configuration panel.
5. Enter a value from 1 – 15 in the BootP Relay Maximum Hop field.
6. Click the Apply button to save the change to the device's running-config file.
7. 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.

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 12-19.

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 12-104.

To configure an IP address and specify the default gateway, use the following CLI method.

USING THE CLI

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>

USING THE WEB MANAGEMENT INTERFACE

You cannot perform initial configuration of the management IP address using the Web management interface, but you can change an address you already configured. You also can configure the default gateway. Use the following procedure.

1. Log on to the Layer 2 Switch using a valid user name and password for read-write access. The System configuration panel is displayed.
2. Click on the plus sign next to Configure in the tree view to display the list of configuration options.
3. Click on the IP Address link
4. When the IP address configuration panel appears, enter the IP address in the IP address field.
5. Enter the subnet mask in the Subnet Mask field.
6. Enter the default gateway's IP address in the Default Gateway field.
7. Click the Apply button to save the change to the device's running-config file.
8. 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.

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.

USING THE CLI

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 THE WEB MANAGEMENT INTERFACE

To map a domain name server to multiple IP addresses:

1. Log on to the Layer 2 Switch using a valid user name and password for read-write access. The System configuration panel is displayed.
2. Select the DNS link to display the DNS panel.
3. Enter the domain name in the Domain Name field.
4. Enter an IP address for each device that will serve as a gateway to the domain name server.

   NOTE: The first address entered will be the primary DNS gateway address. The other addresses will be used in chronological order, left to right, if the primary address is available.

5. Click the Apply button to save the change to the device's running-config file.
6. 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.

Using a DNS Name To Initiate a Trace Route

EXAMPLE:

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.

USING THE CLI

```
FastIron# traceroute nyc02
```

Syntax: traceroute <host-ip-addr> [maxttl <value>] [minttl <value>] [numeric] [timeout <value>] [source-ip <ip addr>]

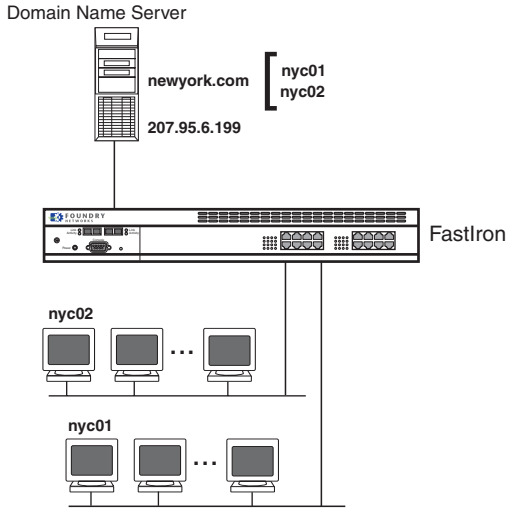
The only required parameter is the IP address of the host at the other end of the route. See the *Foundry Switch and Router Command Line Interface Reference* for information about the parameters.

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 12.12 Querying a host on the newyork.com domain



USING THE WEB MANAGEMENT INTERFACE

1. Log on to the Layer 2 Switch using a valid user name and password for read-only or read-write access. The System configuration panel is displayed.
2. Click on the plus sign next to Command in the tree view to list the command options.
3. Select the Trace Route link to display the Trace Route panel.
4. Enter the host name or IP address in the Target Address field.

   NOTE: You can use the host name only if you have already configured the DNS resolver for the domain that contains the host.

5. Optionally change the minimum and maximum TTLs and the Timeout.
6. Click on Start to begin the trace. The trace results are displayed below the Start and Abort buttons.

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, use the following CLI method.

USING THE CLI

To modify the TTL threshold to 25, enter the following commands:

```
FastIron(config)# ip ttl 25
FastIron(config)# exit
```

Syntax: ip ttl <1-255>

USING THE WEB MANAGEMENT INTERFACE

You cannot change the TTL on a Layer 2 Switch using the Web management interface.

Configuring DHCP Assist

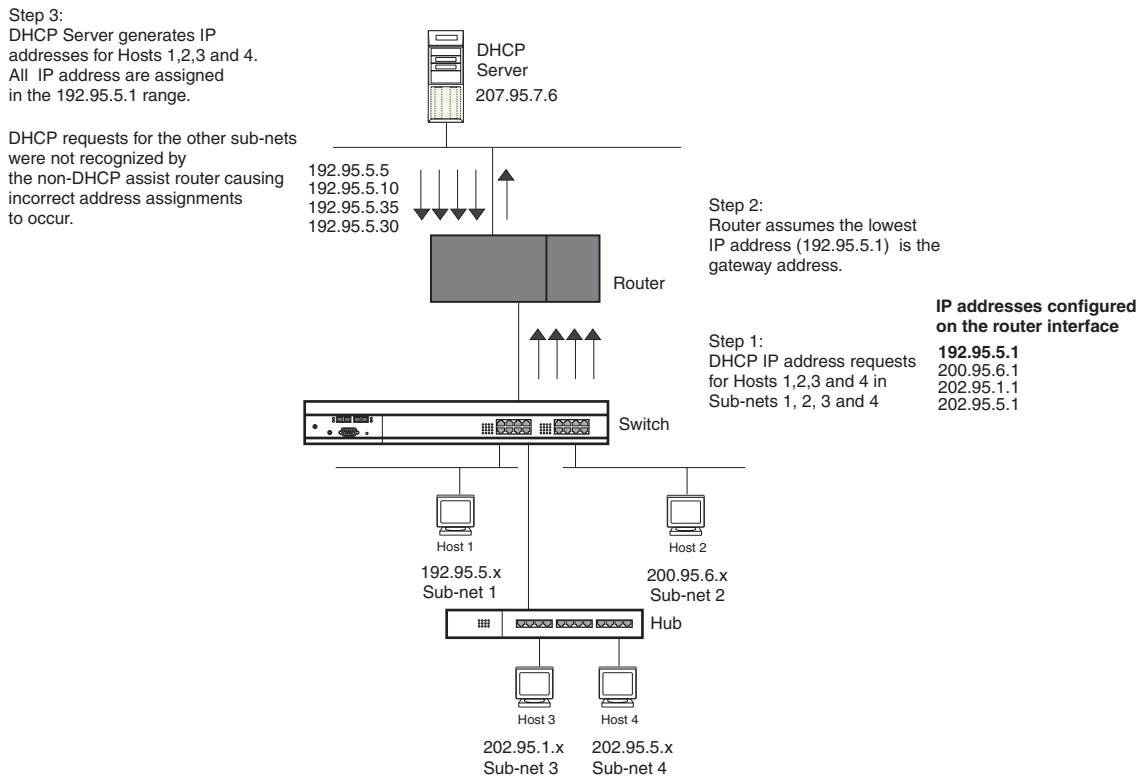
DHCP Assist allows a Foundry Layer 2 Switch to assist a router that is performing multi-netting on its interfaces as part of its DHCP relay function.

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 12-96.

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 12.13 DHCP requests in a network without DHCP Assist on the Layer 2 Switch



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 12.13 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.

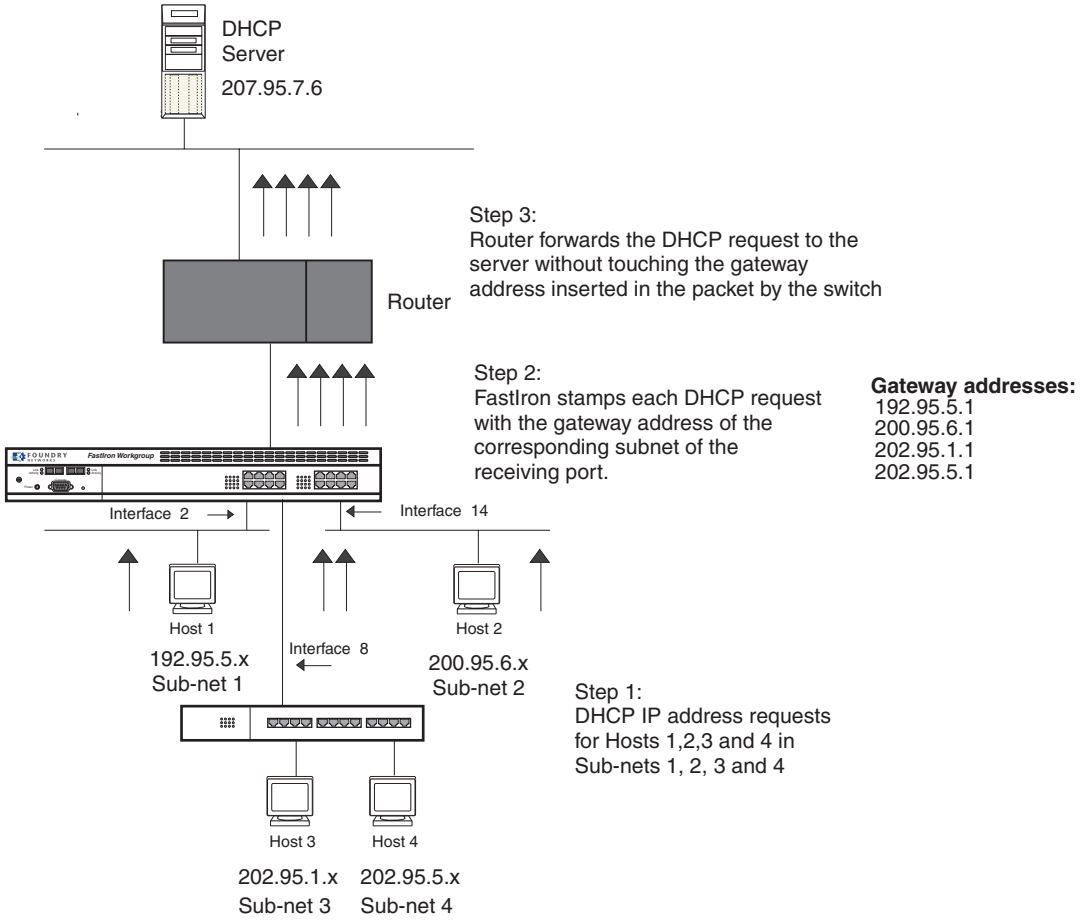
When the DHCP request is received at the server, it assigns all IP addresses within that range only.

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 12.14. 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.

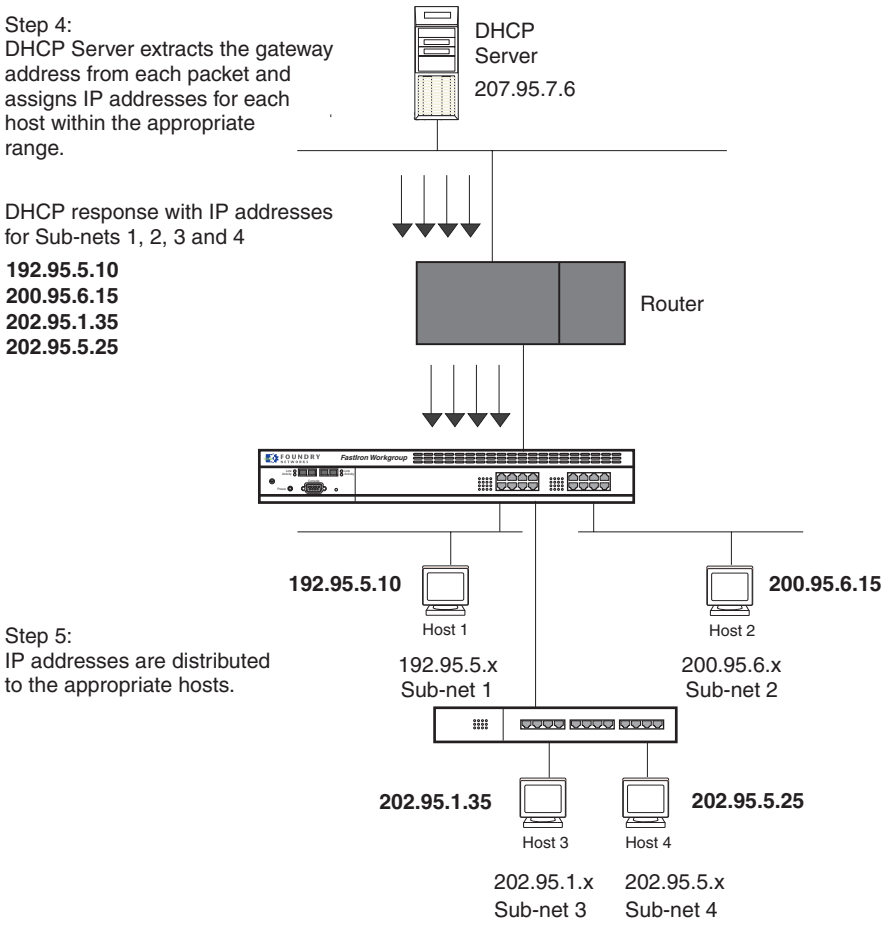
Figure 12.14 DHCP requests in a network with DHCP Assist operating on a FastIron



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 12.15). The IP address is then forwarded back to the workstation that originated the request.

NOTE: The DHCP relay function of the connecting router needs to be turned on.

Figure 12.15 DHCP offers are forwarded back toward the requestors



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.

USING THE CLI

EXAMPLE:

To create the configuration indicated in Figure 12.14 and Figure 12.15:

```
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-2)# dhcp-gateway-list 1
```



```
FastIron(config-if-2)# int e8
FastIron(config-if-8)# dhcp-gateway-list 3
FastIron(config-if-8)# int e 14
FastIron(config-if-14)# dhcp-gateway-list 2
```

Syntax: dhcp-gateway-list <num> <ip-addr>

USING THE WEB MANAGEMENT INTERFACE

1. Log on to the Layer 2 Switch using a valid user name and password for read-write access. The System configuration panel is displayed.
2. Select the DHCP Gateway link to display the DHCP Gateway configuration panel.
3. Enter the list ID in the List ID field. You can specify a number from 1 – 32.
4. Enter up to eight gateway IP address in the IP address fields.
5. Click the Add button to save the change to the device's running-config file.
6. Select the Save link at the bottom of the dialog, then select Yes when prompted to save the configuration change to the startup-config file on the device's flash memory.

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 12-105.
- To display IP information on a Layer 2 Switch, see "Displaying IP Information – Layer 2 Switches" on page 12-129.

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.

USING THE CLI

To enable CIDR format for displaying network masks, entering the following command at the global CONFIG level of the CLI:

```
BigIron(config)# ip show-subnet-length
```

Syntax: [no] ip show-subnet-length

USING THE WEB MANAGEMENT INTERFACE

You cannot configure this option using the Web management interface.

Displaying Information for Jumbo Packets

In release 08.0.00, the output of the **show interfaces ethernet** and **show statistics ethernet** command has been enhanced to display information about the number of packets received that were longer than 1518 octets. In the following examples, the new output is highlighted in bold.

```
BigIron# show interfaces ethernet e 1/1
FastEthernet1/1 is up, line protocol is up
  Hardware is FastEthernet, address is 0004.8085.c500 (bia 0004.8085.c500)
  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
  Internet address is 10.1.1.4/24, MTU 1518 bytes, encapsulation ethernet
  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 filtered, 0 runts, 0 giants, DMA received 0 packets, 0 jumbos
  4 packets output, 256 bytes, 0 underruns
  Transmitted 4 broadcasts, 0 multicasts, 0 unicasts
  0 output errors, 0 collisions, DMA transmitted 4 packets, 0 jumbos
```

```
BigIron# show statistics ethernet e 1/25
Port  Link State     Dupl Speed Trunk Tag Priori MAC            Name
1/1   Up   Forward   Full 100M  None  No  level0 0004.8085.c500

 Port 1/1 Counters:
        InOctets            0            OutOctets            256
          InPkts            0              OutPkts              4
       DMA recvd            0            DMA xmited             4
  InBroadcastPkts           0      OutBroadcastPkts             4
  InMulticastPkts           0      OutMulticastPkts             0
    InUnicastPkts           0        OutUnicastPkts             0
      InJumboPkts           0          OutJumboPkts             0
       InDiscards           0           OutDiscards             0
         InErrors           0             OutErrors             0
     InCollisions           0         OutCollisions             0
                                  OutLateCollisions             0
        Alignment           0                  FCS              0
        GiantPkts           0            ShortPkts              0
     InBitsPerSec           0         OutBitsPerSec             0
     InPktsPerSec           0         OutPktsPerSec             0
    InUtilization           0.00%    OutUtilization          0.00
```

Syntax: show statistics ethernet

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 12-106.

- CPU utilization statistics – see “Displaying CPU Utilization Statistics” on page 12-108.
- IP interfaces – see “Displaying IP Interface Information” on page 12-110.
- ARP entries – see “Displaying ARP Entries” on page 12-113.
- Static ARP entries – see “Displaying ARP Entries” on page 12-113.
- IP forwarding cache – see “Displaying the Forwarding Cache” on page 12-116.
- IP route table – see “Displaying the IP Route Table” on page 12-118.
- IP traffic statistics – see “Displaying IP Traffic Statistics” on page 12-122.

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 information – see “Displaying RIP Filters” on page 13-17.
- OSPF information – see “Displaying OSPF Information” on page 15-49.
- BGP4 information – see “Displaying BGP4 Information” on page 16-111.
- DVMRP information – see the “Show Commands” chapter in the *Foundry Switch and Router Command Line Interface Reference*.
- PIM information – see the “Show Commands” chapter in the *Foundry Switch and Router Command Line Interface Reference*.
- VRRP or VRRPE information – see “Displaying VRRP and VRRPE Information” on page 19-20.
- FSRP information – see the “Show Commands” chapter in the *Foundry Switch and Router Command Line Interface Reference*.

Displaying Global IP Configuration Information

To display global IP configuration information for the router, use one of the following methods.

USING THE CLI

To display IP configuration information, enter the following command at any CLI level:

```
BigIron> 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 12.12: 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 12-50.
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 12-45.
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 12-96.
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 12-40.
enabled	The IP-related protocols that are enabled on the router.
disabled	The IP-related protocols that are disabled on the router.
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 16-38.
Policies	

Table 12.12: CLI Display of Global IP Configuration Information – Layer 3 Switch (Continued)

This Field...	Displays...
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.

USING THE WEB MANAGEMENT INTERFACE

You cannot display global IP configuration information using the Web management interface.

Displaying CPU Utilization Statistics

You can display CPU utilization statistics for IP protocols using the **show process cpu** command.

Beginning with software release 07.6.02, the **show process cpu** command includes CPU utilization statistics for ACL, NAT, 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.

USING THE CLI

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:

```
BigIron# 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
NAT              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            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:

```
BigIron# 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
NAT              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            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:

```
BigIron# 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
NAT                0      0.00
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.

USING THE WEB MANAGEMENT INTERFACE

You cannot display this information using the Web management interface.

Displaying IP Interface Information

To display IP interface information, use one of the following methods.

USING THE CLI

To display IP interface information, enter the following command at any CLI level:

```
BigIron(config)# 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 <portnum>] | [loopback <num>] | [ve <num>]

This display shows the following information.

Table 12.13: CLI Display of Interface IP Configuration Information

This Field...	Displays...
Interface	The type and the slot and port number of the interface.

Table 12.13: CLI Display of Interface IP Configuration Information (Continued)

This Field...	Displays...
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.
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:

```
BigIron# 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
```

USING THE WEB MANAGEMENT INTERFACE

To display IP interface information:

1. Log on to the Layer 3 Switch using a valid user name and password for read-only or read-write access. The System configuration panel is displayed.
2. Click on the plus sign next to Configure in the tree view.
3. Click on the plus sign next to IP in the tree view to expand the list of IP option links.
4. Click on the Interface link to display the IP interface table.

This display shows the following information.

Table 12.14: Web Display of IP Interface Information

This Field...	Displays...
Port #	The physical port number or virtual interface (VE) number. VEs are shown as “v<num>”, where <num> is the number you assigned to the VE when you configured it. For example, VE 1 is shown as “v1”. If a range of ports is listed in this field, the interface is a trunk group. If two ranges of ports are listed, the interface is a trunk group that spans multiple chassis modules.
Encapsulation	The frame type used to encapsulate packets on this interface. The frame type is always Ethernet II.
MTU	The Maximum Transmission Unit (MTU), which specifies the maximum packet size for packets sent and received on this interface.
Metric	The cost associated with this interface.
Directed Broadcast Forward	The state of the directed broadcast forwarding feature. The state can be one of the following: • Disable • Enable To change the state of this feature, see “Enabling Forwarding of Directed Broadcasts” on page 12-50.

Displaying Interface Name in Syslog

By default an interface’s slot number (if applicable) and port number are displayed when you display Syslog messages. Beginning with release 07.6.02, you can display the name of the interface instead of its number by entering a command such as the following:

```
BigIron(config)# ip show-portname
```

This command is applied globally to all interfaces on Layer 2 Switches and Layer 3 Switches.

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:

```
BigIron># 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 ARP cache, use one of the following methods.

USING THE CLI

To display the contents of the ARP cache, enter the following command at any CLI level:

```
BigIron# show arp

Total number of ARP entries: 57    IP Address        MAC Address       Type
Age        Port
1     207.95.6.102        0800.5afc.ea21      Dynamic     0         1/13
2     207.95.6.18         00a0.24d2.04ed      Dynamic     3         1/15
3     207.95.6.54         00a0.24ab.cd2b      Dynamic     0         1/7
4     207.95.6.101        0800.207c.a7fa      Dynamic     0         1/8
5     207.95.6.211        00c0.2638.ac9c      Dynamic     0         1/9
6     1.1.21.2            0004.809e.2e15      Static      None      1/21
6     1.1.21.2            0004.809e.2e15      Static      None      1/22
```

Syntax: show arp [ethernet <portnum> | mac-address <xxxx.xxxx.xxxx> [<mask>] | <ip-addr> [<ip-mask>]] [<num>]

The **ethernet** <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 12.15: CLI Display of ARP Cache

This Field...	Displays...
(route number)	ID of the entry. Beginning with Enterprise release 08.0.00, if multiple outgoing ports for a static route is configured, there will be more than one entry for that route as in route #6 above.
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 12-106. To change the ARP aging interval, see “Changing the ARP Aging Period” on page 12-45. **Note**: Static entries do not age out.
Port	The port on which the entry was learned.

USING THE WEB MANAGEMENT INTERFACE

To display the IP ARP cache:

1. Log on to the Layer 3 Switch using a valid user name and password for read-only or read-write access. The System configuration panel is displayed.
2. Click on the plus sign next to Monitor in the tree view to list the monitoring options.
3. Click on the ARP Cache link to display the IP ARP cache.

This display shows the following information.

Table 12.16: Web Display of ARP Cache – Layer 3 Switch

This Field...	Displays...
Node	The IP address of the device.
MAC Address	The MAC address of the device.

Table 12.16: Web Display of ARP Cache – Layer 3 Switch (Continued)

This Field...	Displays...
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 cache. To display the ARP aging period, see "Displaying Global IP Configuration Information" on page 12-106. To change the ARP aging interval, see "Changing the ARP Aging Period" on page 12-45. **Note**: Static entries do not age out.
Port	The port attached to the device the entry is for. For dynamic entries, this is the port on which the entry was learned.

Displaying the Static ARP Table

To display the static ARP table instead of the ARP cache, use either of the following methods.

USING THE CLI

To display the static ARP table, enter the following command at any CLI level:

```
BigIron# 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 <portnum> | mac-address <xxxx.xxxx.xxxx> [<mask>] | <ip-addr> [<ip-mask>]] [<num>]

The **ethernet** <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 12.17: 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 12-48.
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.

USING THE WEB MANAGEMENT INTERFACE

You cannot display the static ARP table using the Web management interface.

Displaying the Forwarding Cache

To display the IP forwarding cache, use one of the following methods.

NOTE: To display only the forwarding cache entries for aggregated default network routes, see “CAM Default Route Aggregation” on page 12-85.

USING THE CLI

To display the IP forwarding cache, enter the following command at any CLI level:

```
BigIron> 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 12.18: CLI Display of IP Forwarding Cache – Layer 3 Switch

This Field...	Displays...
IP Address	The IP address of the destination.
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.

USING THE WEB MANAGEMENT INTERFACE

To display the IP forwarding cache:

1. Log on to the Layer 3 Switch using a valid user name and password for read-only or read-write access. The System configuration panel is displayed.
2. Click on the plus sign next to Monitor in the tree view to list the monitoring options.
3. Click on the plus sign next to IP to list the IP monitoring options.
4. Click on the Cache link to display the IP cache.

This display shows the following information.

Table 12.19: Web Display of IP Forwarding Cache Information – Layer 3 Switch

This Field...	Displays...
IP Address	The IP address of the destination.
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 • I – ICMP Deny • P – Permanent • F – Forward • U – Us • C – Complex Filter • K – Drop • W – Wait ARP • R – Fragment • S – Snap Encap
Action	This information is used by Foundry customer support.
Flag Check	This information is used by Foundry customer support.
Snap	This information is used by Foundry customer support.
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.
Priority	The QoS priority of the port or VLAN.

Displaying the IP Route Table

To display the IP route table, use one of the following methods.

USING THE CLI

To display the IP route table, enter the following command at any CLI level:

```
BigIron> 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    1.1.0.0           255.255.0.0       99.1.1.2          1/1    2      R
2    1.2.0.0           255.255.0.0       99.1.1.2          1/1    2      R
3    1.3.0.0           255.255.0.0       99.1.1.2          1/1    2      R
4    1.4.0.0           255.255.0.0       99.1.1.2          1/1    2      R
5    1.5.0.0           255.255.0.0       99.1.1.2          1/1    2      R
6    1.6.0.0           255.255.0.0       99.1.1.2          1/1    2      R
7    1.7.0.0           255.255.0.0       99.1.1.2          1/1    2      R
8    1.8.0.0           255.255.0.0       99.1.1.2          1/1    2      R
9    1.9.0.0           255.255.0.0       99.1.1.2          1/1    2      R
10   1.10.0.0          255.255.0.0       99.1.1.2          1/1    2      S
11   200.200.200.2     255.255.255.255   0.0.0.0           1/21   1      SA
```

Syntax: show ip route [<ip-addr> [<ip-mask>] [longer] [none-bgp]] | <num> | bgp | direct | ospf | rip | static | tunnel | summary]

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 **tunnel** option displays only routes that are using an MPLS LSP as a shortcut.

The **summary** option (Service Provider release 09.1.01 and higher) displays a summary of the information in the IP route table.

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:

```
BigIron(config)# 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
 1    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:

```
BigIron(config)# 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
 1    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:

```
BigIron(config)# 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:

```
BigIron# 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 12.20: 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.
	• SA – A static route that has multiple outgoing ports in its entry.

USING THE WEB MANAGEMENT INTERFACE

To display the IP route table:

1. Log on to the Layer 3 Switch using a valid user name and password for read-only or read-write access. The System configuration panel is displayed.
2. Click on the plus sign next to Monitor in the tree view to list the monitoring options.
3. Click on the plus sign next to IP to list the IP monitoring options.
4. Click on the Routing Table link to display the table.

Clearing IP Routes

If needed, you can clear the entire route table or specific individual routes. To do so, use one of the following procedures.

USING THE CLI

To clear all routes from the IP route table:

```
BigIron# clear ip route
```

To clear route 209.157.22.0/24 from the IP routing table:

```
BigIron# 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>]

USING THE WEB MANAGEMENT INTERFACE

The Web management interface does not allow you to selectively clear routes in the IP routing table, but does allow you to clear all routes from the IP routing table.

To clear all routes from the IP route table:

1. Log on to the Layer 3 Switch using a valid user name and password for read-write access. The System configuration panel is displayed.
2. Click on the plus sign next to Command in the tree view to expand the list of command options.
3. Click on the Clear link to display the Clear panel.
4. Select the box next to IP Route.
5. Click Apply.

Displaying IP Traffic Statistics

To display IP traffic statistics, use one of the following methods.

USING THE CLI

To display IP traffic statistics, enter the following command at any CLI level:

```
BigIron> 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 12.21: 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.

Table 12.21: CLI Display of IP Traffic Statistics – Layer 3 Switch (Continued)

This Field...	Displays...
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 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.

Table 12.21: CLI Display of IP Traffic Statistics – Layer 3 Switch (Continued)

This Field...	Displays...
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”.	
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.
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 12.21: 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 this router 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.

USING THE WEB MANAGEMENT INTERFACE

To display IP traffic statistics:

1. Log on to the Layer 3 Switch using a valid user name and password for read-only or read-write access. The System configuration panel is displayed.
2. Click on the plus sign next to Monitor in the tree view to list the monitoring options.
3. Click on the plus sign next to IP to list the IP monitoring options.
4. Click on the Traffic link to display the table.

This display shows the following information.

Table 12.22: Web Display of IP Traffic Statistics – Layer 3 Switch

This Field...	Displays...
IP statistics	
Packets Received	The number of IP packets received by the device.
Packets Sent	The number of IP packets originated and sent by the device.
Packets Forwarded	The number of IP packets received from another device and forwarded by this device.
Filtered	The number of IP packets filtered by this device.
Fragmented	The number of IP packets fragmented by this device before sending or forwarding them.
Reassembled	The number of fragmented IP packets received and re-assembled by the device.
Bad Header	The number of packets dropped because they had a bad header.
No Route	The number of packets dropped because they had no route information.

Table 12.22: Web Display of IP Traffic Statistics – Layer 3 Switch (Continued)

This Field...	Displays...
Unknown Protocols	The number of packets dropped because they were using an unknown protocol.
No Buffer	The number of packets dropped because the device ran out of buffer space.
Other Errors	The number of packets dropped due to errors other than the ones listed above.
ICMP statistics	
Total Received	The number of ICMP packets received by the device.
Total Sent	The number of ICMP packets sent by the device.
Received Errors	This information is used by Foundry customer support.
Sent Errors	This information is used by Foundry customer support.
Received Unreachable	The number of Destination Unreachable messages received by the device.
Sent Unreachable	The number of Destination Unreachable messages sent by the device.
Received Time Exceed	The number of Time Exceeded messages received by the device.
Sent Time Exceed	The number of Time Exceeded messages sent by the device.
Received Parameter	The number of Parameter Problem messages received by the device.
Sent Parameter	The number of Parameter Problem messages sent by the device.
Received Source Quench	The number of Source Quench messages received by the device.
Sent Source Quench	The number of Source Quench messages sent by the device.
Received Redirect	The number of Redirect messages received by the device.
Sent Redirect	The number of Redirect messages sent by the device.
Received Echo	The number of Echo messages received by the device.
Sent Echo	The number of Echo messages sent by the device.
Received Echo Reply	The number of Echo messages received by the device.
Sent Echo Reply	The number of Echo messages sent by the device.
Received Timestamp	The number of Timestamp messages received by the device.
Sent Timestamp	The number of Timestamp messages sent by the device.
Received Timestamp Reply	The number of Timestamp Reply messages received by the device.
Sent Timestamp Reply	The number of Timestamp Reply messages sent by the device.
Received Address Mask	The number of Address Mask Request messages received by the device.
Sent Address Mask	The number of Address Mask Request messages sent by the device.

Table 12.22: Web Display of IP Traffic Statistics – Layer 3 Switch (Continued)

This Field...	Displays...
Received Address Mask Reply	The number of Address Mask Replies messages received by the device.
Sent Address Mask Reply	The number of Address Mask Replies messages sent by the device.
Received IRDP Advertisement	The number of ICMP Router Discovery Protocol (IRDP) Advertisement messages received by the device.
Sent IRDP Advertisement	The number of IRDP Advertisement messages sent by the device.
Received IRDP Solicitation	The number of IRDP Solicitation messages received by the device.
Sent IRDP Solicitation	The number of IRDP Solicitation messages sent 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”.	
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.
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.

Table 12.22: Web Display of IP Traffic Statistics – Layer 3 Switch (Continued)

This Field...	Displays...
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.
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 Address Family	The number of RIP packets dropped because the value in the Address Family Identifier field of the packet's header was invalid.
Bad Request 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 Response Format	The number of responses to RIP request packets this router dropped because the format was bad.
Resp Not From RIP Port	This information is used by Foundry customer support.
Response 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 12-129.
- ARP entries – see "Displaying ARP Entries" on page 12-130.
- IP traffic statistics – see "Displaying IP Traffic Statistics" on page 12-132.

Displaying Global IP Configuration Information

To display the Layer 2 Switch's IP address and default gateway, use either of the following methods.

USING THE CLI

To display the IP configuration, enter the following command from any level of the CLI:

```
FastIron(config)# 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 12.23: CLI Display of Global IP Configuration Information – Layer 2 Switch

This Field...	Displays...
IP configuration	
Switch IP address	The management IP address you 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.
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 Layer 2 Switch has contacted a TFTP server since the last time the software was reloaded or the Layer 2 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.

USING THE WEB MANAGEMENT INTERFACE

To display the management IP address and default gateway:

1. Log on to the Layer 2 Switch using a valid user name and password for read-write access. The System configuration panel is displayed.
2. Click on the plus sign next to Configure in the tree view to display the list of configuration options.
3. Select the IP Address link to display the IP address configuration panel.

NOTE: You cannot display the TFTP access information using the Web management interface.

Displaying ARP Entries

To display the entries the Layer 2 Switch has placed in its ARP cache, use either of the following methods:

USING THE CLI

To display the ARP cache, enter the following command from any level of the CLI:

```
FastIron(config)# 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 12.24: 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.

USING THE WEB MANAGEMENT INTERFACE

To display the ARP cache:

1. Log on to the Layer 2 Switch using a valid user name and password for read-only or read-write access. The System configuration panel is displayed.
2. Click on the plus sign next to Monitor in the tree view to display the list of configuration options.
3. Select the ARP Cache link to display the ARP cache.

This display shows the following information.

Table 12.25: Web Display of ARP Cache – Layer 2 Switch

This Field...	Displays...
Node	The IP address of the device.
MAC Address	The MAC address of the device.

Table 12.25: Web Display of ARP Cache – Layer 2 Switch (Continued)

This Field...	Displays...
Type	The type, which is always Dynamic on Foundry Layer 2 Switches. The device learns dynamic entries from incoming packet.
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.
Port	The port on which the entry was learned.

Displaying IP Traffic Statistics

To display IP traffic statistics on a Layer 2 Switch, use one of the following methods.

USING THE CLI

To display IP traffic statistics, 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 12.26: 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.
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.

Table 12.26: CLI Display of IP Traffic Statistics – Layer 2 Switch (Continued)

This Field...	Displays...
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.
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.

Table 12.26: CLI Display of IP Traffic Statistics – Layer 2 Switch (Continued)

This Field...	Displays...
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.

USING THE WEB MANAGEMENT INTERFACE

To display IP traffic statistics:

1. Log on to the Layer 2 Switch using a valid user name and password for read-only or read-write access. The System configuration panel is displayed.
2. Click on the plus sign next to Monitor in the tree view to list the monitoring options.
3. Click on the plus sign next to IP to list the IP monitoring options.
4. Click on the Traffic link to display the table.

This display shows the following information.

Table 12.27: Web Display of IP Traffic Statistics – Layer 2 Switch

This Field...	Displays...
IP statistics	
Packets Received	The number of IP packets received by the device.
Packets Sent	The number of IP packets originated and sent by the device.
Fragmented	The number of IP packets fragmented by this device before sending or forwarding them.
Reassembled	The number of fragmented IP packets received and re-assembled by the device.
Bad Header	The number of packets dropped because they had a bad header.
No Route	The number of packets dropped because they had no route information.
Unknown Protocols	The number of packets dropped because they were using an unknown protocol.
No Buffer	The number of packets dropped because the device ran out of buffer space.
Other Errors	The number of packets dropped due to errors other than the ones listed above.
ICMP statistics	
Total Received	The number of ICMP packets received by the device.
Total Sent	The number of ICMP packets sent by the device.
Received Errors	This information is used by Foundry customer support.

Table 12.27: Web Display of IP Traffic Statistics – Layer 2 Switch (Continued)

This Field...	Displays...
Sent Errors	This information is used by Foundry customer support.
Received Unreachable	The number of Destination Unreachable messages received by the device.
Sent Unreachable	The number of Destination Unreachable messages sent by the device.
Received Time Exceed	The number of Time Exceeded messages received by the device.
Sent Time Exceed	The number of Time Exceeded messages sent by the device.
Received Parameter	The number of Parameter Problem messages received by the device.
Sent Parameter	The number of Parameter Problem messages sent by the device.
Received Source Quench	The number of Source Quench messages received by the device.
Sent Source Quench	The number of Source Quench messages sent by the device.
Received Redirect	The number of Redirect messages received by the device.
Sent Redirect	The number of Redirect messages sent by the device.
Received Echo	The number of Echo messages received by the device.
Sent Echo	The number of Echo messages sent by the device.
Received Echo Reply	The number of Echo messages received by the device.
Sent Echo Reply	The number of Echo messages sent by the device.
Received Timestamp	The number of Timestamp messages received by the device.
Sent Timestamp	The number of Timestamp messages sent by the device.
Received Timestamp Reply	The number of Timestamp Reply messages received by the device.
Sent Timestamp Reply	The number of Timestamp Reply messages sent by the device.
Received Address Mask	The number of Address Mask Request messages received by the device.
Sent Address Mask	The number of Address Mask Request messages sent by the device.
Received Address Mask Reply	The number of Address Mask Replies messages received by the device.
Sent Address Mask Reply	The number of Address Mask Replies messages sent by the device.
Received IRDP Advertisement	The number of ICMP Router Discovery Protocol (IRDP) Advertisement messages received by the device.
Sent IRDP Advertisement	The number of IRDP Advertisement messages sent by the device.
Received IRDP Solicitation	The number of IRDP Solicitation messages received by the device.
Sent IRDP Solicitation	The number of IRDP Solicitation messages sent by the device.
UDP statistics	
Received	The number of UDP packets received by the device.

Table 12.27: Web Display of IP Traffic Statistics – Layer 2 Switch (Continued)

This Field...	Displays...
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”.	
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.
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.
Keepalive Close Connection	This information is used by Foundry customer support.
Keepalive Failure Callback	This information is used by Foundry customer support.
TCP Connect Connection Exist	This information is used by Foundry customer support.
TCP Connect Out of TCB	This information is used by Foundry customer support.

Chapter 13
Configuring RIP

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 13.1 lists the global RIP parameters and their default values, and indicates where you can find configuration information.

Table 13.1: 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 13.2 on page 13-3.	Disabled	13-3
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	13-6
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	13-6
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)	13-9
Update interval	How often the router sends route updates to its RIP neighbors	30 seconds	13-10
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 13.2 on page 13-3.	Disabled	13-11
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	13-12

RIP Interface Parameters

Table 13.2 lists the interface-level RIP parameters and their default values, and indicates where you can find configuration information.

Table 13.2: 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	13-3
Metric	A numeric cost the router adds to RIP routes learned on the interface. This parameter applies only to RIP routes.	1 (one)	13-4
Learning default routes	Locally overrides the global setting. See Table 13.1 on page 13-2.	Disabled	13-11
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.	13-13
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.	13-14

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 one of the following methods.

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.

USING THE CLI

To enable RIP globally, enter the following command:

```
BigIron(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:

```
BigIron(config)# interface ethernet 1/1
BigIron(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.

USING THE WEB MANAGEMENT INTERFACE

After globally enabling the protocol, you must enable it on individual interfaces. To enable RIP globally:

1. Log on to the device using a valid user name and password for read-write access. The System configuration dialog is displayed.
2. Select the Enable radio button next to RIP.
3. Click the Apply button to apply the changes to the device's running-config.
4. Click 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.

To enable RIP on an individual 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 RIP in the tree view to expand the list of RIP option links.
3. Click on the Interface link to display the RIP interface table.
4. Click on the Modify button in the row for the port.
5. Select the RIP version from the pulldown menu. The default is version 2.
6. Click the Apply button to save the change to the device's running-config.

   NOTE: To apply the changes to all RIP interfaces, select the Apply To All Ports button instead of the Apply button.

7. To configure settings for another interface, select the port (and slot, if applicable) from the top of the panel, then go to Step 5.
8. 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.

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 either of the following methods.

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.

USING THE CLI

To increase the cost a port adds to RIP routes learned in that port, enter commands such as the following:

```
BigIron(config)# interface ethernet 6/1
BigIron(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>

USING THE WEB MANAGEMENT INTERFACE

1. Log on to the device using a valid user name and password for read-write access. The System configuration panel is displayed.
2. Click on the plus sign next to Configure in the tree view to expand the list of configuration options.
3. Click on the plus sign next to IP in the tree view to expand the list of IP option links.
4. Click on the Interface link to display the interface table.
5. Click on the Modify button in the row for the port.
6. Enter a value from 1 – 16 for the metric.
7. Click the Add button to save the change to the device's running-config file.
8. To configure settings for another port, select the port (and slot, if applicable) and go to step 6.
9. 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.

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.

USING THE CLI

To configure a global RIP offset list, enter commands such as the following:

```
BigIron(config)# access-list 21 deny 160.1.0.0 0.0.255.255
BigIron(config)# access-list 21 permit any
BigIron(config)# router rip
BigIron(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 | pos <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.

```
BigIron(config-rip-router)# offset-list 21 in ethernet 2/1
```

USING THE WEB MANAGEMENT INTERFACE

You cannot configure this option using the Web management interface.

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 16-38 for a list of the default distances for all route sources.

USING THE CLI

To change the administrative distance for RIP routes, enter a command such as the following:

```
BigIron(config-rip-router)# distance 140
```

This command changes the administrative distance to 140 for all RIP routes.

Syntax: [no] distance <num>

USING THE WEB MANAGEMENT INTERFACE

1. Log on to the device using a valid user name and password for read-write access. The System configuration panel is displayed.
2. Click on the plus sign next to Configure in the tree view to expand the list of configuration options.
3. Click on the plus sign next to RIP in the tree view to expand the list of RIP option links.
4. Click on the General link to display the RIP configuration panel, shown in Figure 13.1 on page 13-11.
5. Edit the value in the Distance field.
6. Click the Apply button to save the change to the device's running-config file.
7. To configure settings for another port, select the port (and slot, if applicable) and go to step 5.
8. 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.

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 permits 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.

USING THE CLI

To configure a redistribution filter, enter a command such as the following:

```
BigIron(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:

```
BigIron(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:

```
BigIron(config-rip-router)# deny redistribute 3 ospf address 207.92.0.0 255.255.0.0
match-metric 10
```

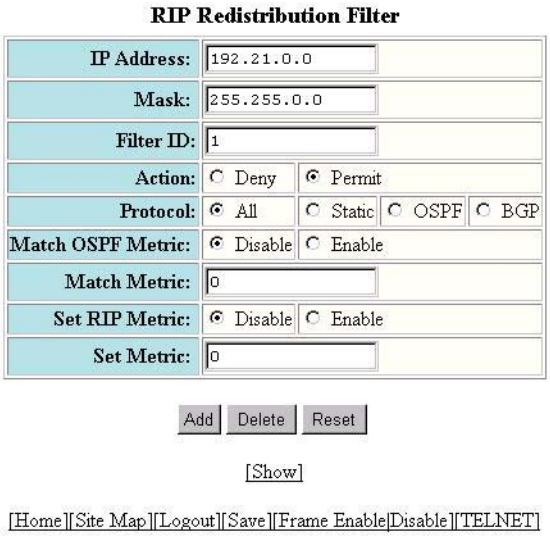

The following commands deny redistribution of all routes except routes for 10.10.10.x and 20.20.20.x:

```
BigIron(config-rip-router)# deny redistribute 64 static address 255.255.255.255
255.255.255.255
BigIron(config-rip-router)# permit redistribute 1 static address 10.10.10.0
255.255.255.0
BigIron(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.

USING THE WEB MANAGEMENT INTERFACE

1. Log on to the device using a valid user name and password for read-write access. The System configuration panel is displayed.
2. Click on the plus sign next to Configure in the tree view to expand the list of configuration options.
3. Click on the plus sign next to RIP in the tree view to expand the list of RIP option links.
4. Click on the Redistribution Filter link.
   - If the device does not have any RIP redistribution filters, the RIP Redistribution Filter configuration panel is displayed, as shown in the following example.
   - If a RIP redistribution filter is already configured and you are adding a new filter, click on the Add Redistribution Filter link to display the RIP Neighbor Filter configuration panel, as shown in the following example.
   - If you are modifying an existing RIP redistribution filter, click on the Modify button to the right of the row describing the filter to display the RIP Redistribution Filter configuration panel, as shown in the following example.



5. Enter an IP address and mask to filter on a specific network. You can use zeros (0.0.0.0) instead of a specific interface to allow all IP addresses or mask ranges.
6. Enter the filter ID.
7. Select either Permit or Deny as the action.
8. Select the types of routes you want to filter on next to Protocol.

9. Enable the Match Metric parameter if you want to limit the import of routes to only those that match the metric specified in the Match Metric field.
10. Enable the Set Metric parameter to define and assign a specific metric to an imported route. If enabled, the specified value overrides the default metric defined on the RIP configuration panel.
11. Click the Add button to save the change to the device's running-config file.
12. 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.

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.

USING THE CLI

To change the RIP metric the Layer 3 Switch assigns to redistributed routes, enter a command such as the following:

```
BigIron(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>

USING THE WEB MANAGEMENT INTERFACE

1. Log on to the device using a valid user name and password for read-write access. The System configuration panel is displayed.
2. Click on the plus sign next to Configure in the tree view to expand the list of configuration options.
3. Click on the plus sign next to RIP in the tree view to expand the list of RIP option links.
4. Click on the General link to display the RIP configuration panel, shown in Figure 13.1 on page 13-11.
5. Enter a value from 1 – 15 in the Redistribution Default Metric field.
6. Click the Apply button to save the change to the device's running-config file.
7. To configure settings for another port, select the port (and slot, if applicable) and go to step 5.
8. 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.

Enabling Redistribution

After you configure redistribution parameters, you need to enable redistribution.

USING THE CLI

To enable RIP redistribution, enter the following command:

```
BigIron(config-rip-router)# redistribution
```

Syntax: [no] redistribution

USING THE WEB MANAGEMENT INTERFACE

1. Log on to the device using a valid user name and password for read-write access. The System configuration panel is displayed.
2. Click on the plus sign next to Configure in the tree view to expand the list of configuration options.
3. Click on the plus sign next to RIP in the tree view to expand the list of RIP option links.
4. Click on the General link to display the RIP configuration panel, shown in Figure 13.1 on page 13-11.
5. Select Disable or Enable next to Redistribution.

6. Click the Apply button to save the change to the device's running-config file.
7. To configure settings for another port, select the port (and slot, if applicable) and go to step 5.
8. 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.

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.

USING THE CLI

To change the RIP update interval, enter a command such as the following:

```
BigIron(config-rip-router)# update 120
```

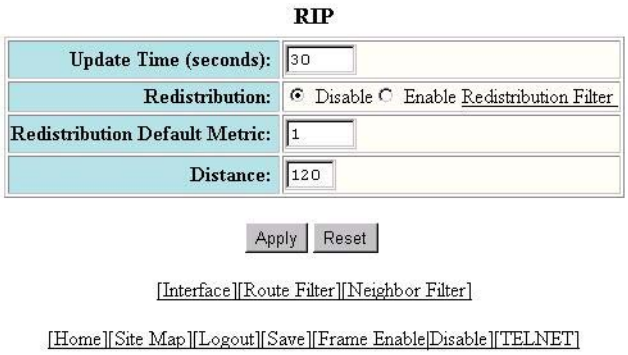
This command configures the Layer 3 Switch to send RIP updates every 120 seconds.

Syntax: update-time <1-1000>

USING THE WEB MANAGEMENT INTERFACE

1. Log on to the device using a valid user name and password for read-write access. The System configuration panel is displayed.
2. Click on the plus sign next to Configure in the tree view to expand the list of configuration options.
3. Click on the plus sign next to RIP in the tree view to expand the list of RIP option links.
4. Click on the General link to display the RIP configuration panel, shown in Figure 13.1 on page 13-11.
5. Enter a value from 1 – 1000 in the Update Time field.
6. Click the Apply button to save the change to the device's running-config file.
7. To configure settings for another port, select the port (and slot, if applicable) and go to step 5.
8. 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.

Figure 13.1 RIP configuration panel



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.

USING THE CLI

To enable learning of default RIP routes on a global basis, enter the following command:

```
BigIron(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:

```
BigIron(config)# interface ethernet 1/1
BigIron(config-if-1/1)# ip rip learn-default
```

Syntax: [no] ip rip learn-default

USING THE WEB MANAGEMENT INTERFACE

To enable learning of default RIP routes:

1. Log on to the device using a valid user name and password for read-write access. The System configuration panel is displayed.
2. Click on the plus sign next to Configure in the tree view to expand the list of configuration options.
3. Click on the plus sign next to RIP in the tree view to expand the list of RIP option links.
4. Click on the Interface link to display the RIP interface table.
5. Click on the Modify button in the row for the port.
6. Select Disable or Enable next to Learn Default.
7. Click the Apply button to save the change to the device's running-config file.

NOTE: To apply the changes to all RIP interfaces, select the Apply To All Ports button instead of the Apply button.

---

8. To configure settings for another port, select the port (and slot, if applicable) at the top of the panel, then go to step 5.
9. 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.

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.

USING THE CLI

To configure a RIP neighbor filters, enter a command such as the following:

```
BigIron(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. Thus, to deny learning from a specific neighbor but allow all other neighbors, you must add a filter that allows learning from all neighbors. Make sure you add the filter to permit all neighbors as the last filter (the one with the highest filter number). Otherwise, the software can match on the permit all filter before a filter that denies a specific neighbor, and learn routes from that neighbor.

```
BigIron(config-rip-router)# neighbor 2 deny 192.16.1.170
BigIron(config-rip-router)# neighbor 1024 permit any
```

USING THE WEB MANAGEMENT INTERFACE

To define a RIP neighbor filter:

1. Log on to the device using a valid user name and password for read-write access. The System configuration panel is displayed.
2. Click on the plus sign next to Configure in the tree view to expand the list of configuration options.
3. Click on the plus sign next to RIP in the tree view to expand the list of RIP option links.
4. Click on the Neighbor Filter link.
   - If the device does not have any RIP neighbor filters, the RIP Neighbor Filter configuration panel is displayed, as shown in the following example.
   - If a RIP neighbor filter is already configured and you are adding a new filter, click on the Add Neighbor Filter link to display the RIP Neighbor Filter configuration panel, as shown in the following example.
   - If you are modifying an existing RIP neighbor filter, click on the Modify button to the right of the row describing the filter to display the RIP Neighbor Filter configuration panel, as shown in the following example.

5. Enter the filter ID.
6. Select either Permit or Deny as the action.

7. Enter the IP address of the RIP neighbor router.
8. Click the Add button to save the change to the device's running-config file.
9. 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.

To modify or delete a RIP neighbor filter:

1. Log on to the device using a valid user name and password for read-write access. The System configuration dialog is displayed.
2. Click on the plus sign next to Configure in the tree view to expand the list of configuration options.
3. Click on the plus sign next to RIP in the tree view to expand the list of RIP option links.
4. Click on the Neighbor Filter link.
5. Click the Modify or Delete button next to the filter that is to be changed or deleted. If you click Modify, enter the changes to the Action or IP Address fields and then click the Modify button apply the changes. If you click Delete, the filter is removed immediately.
6. Click the Add button to save the change to the device's running-config file.
7. 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.

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. Thus, if you disable one method, the other method is enabled.

NOTE: These methods are in addition to RIP's maximum valid route cost of 15.

USING THE CLI

To disable poison reverse and enable split horizon on an interface, enter commands such as the following:

```
BigIron(config)# interface ethernet 1/1
BigIron(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:

```
BigIron(config)# interface ethernet 1/1
BigIron(config-if-1/1)# ip rip poison-reverse
```

USING THE WEB MANAGEMENT INTERFACE

To change the loop prevention method on an interface:

1. Log on to the device using a valid user name and password for read-write access. The System configuration panel is displayed.
2. Click on the plus sign next to Configure in the tree view to expand the list of configuration options.
3. Click on the plus sign next to RIP in the tree view to expand the list of RIP option links.
4. Click on the Interface link to display the RIP interface table.
5. Click on the Modify button in the row for the port.

6. Select Enable or Disable next to Poison Reverse. (Enable enables poison reverse and disables split horizon. Disable enables split horizon and disables poison reverse.)
7. Click the Apply button to save the change to the device's running-config.

NOTE: To apply the changes to all RIP interfaces, select the Apply To All Ports button instead of the Apply button.

8. To configure settings for another port, select the port (and slot, if applicable) at the top of the panel, then go to step 6.
9. 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.

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 19-1.

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.

USING THE CLI

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: [no] use-vrrp-path

The syntax is the same for VRRP and VRRPE.

USING THE WEB MANAGEMENT INTERFACE

See "Configuration Examples" on page 19-35.

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.

USING THE CLI

To configure RIP filters, enter commands such as the following:

```
BigIron(config-rip-router)# filter 1 permit 192.53.4.1 255.255.255.0
BigIron(config-rip-router)# filter 2 permit 192.53.5.1 255.255.255.0
BigIron(config-rip-router)# filter 3 permit 192.53.6.1 255.255.255.0
BigIron(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.

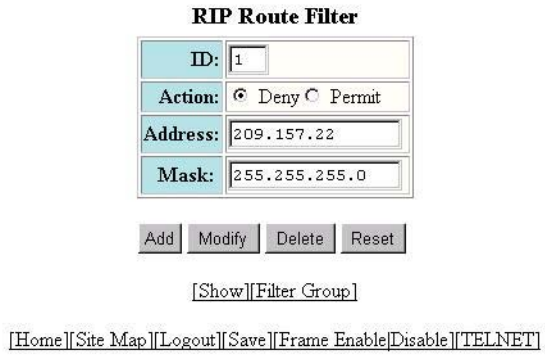
Since the default action is permit, all other routes (routes not explicitly permitted or denied by the filters) can be learned or advertised.

Syntax: filter <filter-num> permit | deny <source-ip-address> | any <source-mask> | any [log]

USING THE WEB MANAGEMENT INTERFACE

To define a RIP route filter:

1. Log on to the device using a valid user name and password for read-write access. The System configuration panel is displayed.
2. Click on the plus sign next to Configure in the tree view to expand the list of configuration options.
3. Click on the plus sign next to RIP in the tree view to expand the list of RIP option links.
4. Click on the Route Filter link.
   - If the device does not have any RIP route filters, the RIP Route Filter configuration panel is displayed, as shown in the following example.
   - If a RIP route filter is already configured and you are adding a new filter, click on the Add Route Filter link to display the RIP Route Filter configuration panel, as shown in the following example.
   - If you are modifying an existing RIP route filter, click on the Modify button to the right of the row describing the filter to display the RIP Route Filter configuration panel, as shown in the following example.



5. Enter the filter ID.
6. Select either Permit or Deny as the action.
7. Enter an IP address and mask or the wildcard value, 0.0.0.0, to allow all routes.
8. Click the Add button to save the change to the device's running-config file.
9. 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.

To modify or delete a RIP route filter:

1. Log on to the device using a valid user name and password for read-write access. The System configuration panel is displayed.
2. Click on the plus sign next to Configure in the tree view to expand the list of configuration options.
3. Click on the plus sign next to RIP in the tree view to expand the list of RIP option links.
4. Select the Route Filter link.
5. Click on the Modify button or Delete button to the right of the row describing the filter.
6. If you are modifying a filter, see the procedure above for configuration information.

7. 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.

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.

USING THE CLI

To apply RIP route filters to an interface, enter commands such as the following:

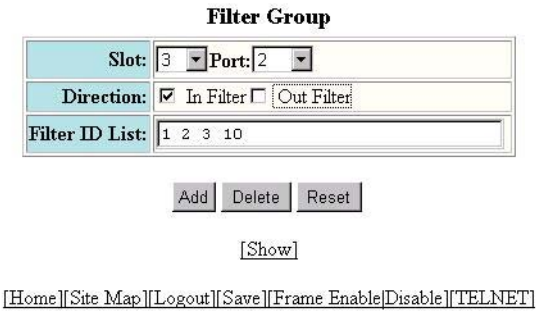
```
BigIron(config)# interface ethernet 1/2
BigIron(config-if-1/2)# ip rip filter-group in 2 3 4
```

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.

USING THE WEB MANAGEMENT INTERFACE

1. Log on to the device using a valid user name and password for read-write access. The System configuration panel is displayed.
2. Click on the plus sign next to Configure in the tree view to expand the list of configuration options.
3. Click on the plus sign next to RIP in the tree view to expand the list of RIP option links.
4. Select the Route Filter link.
5. Select the Filter Group link.
   - If the device does not have any RIP filter groups, the Filter Group configuration panel is displayed, as shown in the following example.
   - If a RIP filter group is already configured and you are adding a new group, click on the Add RIP Route Filter Group link to display the Filter Group configuration panel, as shown in the following example.
   - If you are modifying an existing RIP filter group, click on the Modify button to the right of the row describing the group to display the Filter Group configuration panel, as shown in the following example.



6. Select the port (and slot if applicable) to which you are assigning the filter.
7. Select either or both the In Filter and Out Filter options.
   - Selecting In Filter applies the filters to all RIP updates received on the port.
   - Selecting Out Filter applies the filters to all routes advertised on the port.
   - Selecting both options applies the filters to both incoming updates and outgoing advertisements.

8. Click the Add button to save the change to the device's running-config file.
9. 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.

Setting RIP Timers

In Enterprise software release 07.8.01, you can set three new timers for the RIP protocol. The new **timers-basic** command allows you to set the RIP update timer, aging timeout interval, and garbage-collection timer. The RIP protocol must be enabled on the Foundry device in order to set these timers.

The RIP **update-time** command, available in previous releases, has lower priority than the **timers-basic** command. If both commands are configured on the device, then the **update-time** command is ignored.

For example, the following command sets the three RIP timers:

```
BigIron(config) router rip
BigIron(config-rip-router)# timers-basic 5 15 15
```

Syntax: [no] timers-basic <update-timer> <aging-timeout-interval> <garbage-collection-timer>

The <update-timer> specifies how often RIP update messages are sent. You can specify from 1 – 1,000 seconds. The default is 30 seconds.

The <aging-timeout-interval> specifies how long the Foundry device waits for a route update before declaring a route invalid. The value specified for the <aging-timeout-interval> should be at least three times the value specified for the <update-timer>. The <aging-timeout-interval> can be from 3 – 3,000 seconds. The default is 180 seconds.

The <garbage-collection-timer> specifies how long the Foundry device waits for a route update before removing the route from the RIP route table. The value specified for the <garbage-collection-timer> should be at least three times the value specified for the <update-timer>. The <garbage-collection-timer> can be from 3 – 3,000 seconds. The default is 120 seconds.

Displaying RIP Filters

To display the RIP filters configured on the router, use one of the following methods.

USING THE CLI

To display RIP filters, enter the following command at any CLI level:

```
BigIron> 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 13.3: 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.
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.

Table 13.3: CLI Display of RIP Filter Information (Continued)

This Field...	Displays...
Neighbor IP Address	The IP address of the RIP neighbor.

USING THE WEB MANAGEMENT INTERFACE

To display RIP filter information:

1. Log on to the device using a valid user name and password for read-only or read-write access. The System configuration panel is displayed.
2. Click on the plus sign next to Configure in the tree view.
3. Click on the plus sign next to RIP.
4. Select one of the following links:
   - Neighbor Filter
   - Route Filter
   - Redistribution Filter

Displaying CPU Utilization Statistics

You can display CPU utilization statistics for RIP and other IP protocols.

USING THE CLI

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:

```
BigIron# 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:

```
BigIron# 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:

```
BigIron# 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.

USING THE WEB MANAGEMENT INTERFACE

You cannot display this information using the Web management interface.

Chapter 14
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 of the multicast protocols uses IGMP. IGMP is automatically enabled on an interface when you configure PIM or DVMRP on an interface and is disabled on the interface if you disable PIM or DVMRP on the interface.

NOTE: This chapter applies only to IP multicast routing. To configure Layer 2 IP multicast features, see the "Configuring IP Multicast Traffic Reduction" chapter in the *Foundry Switch and Router Installation and Basic Configuration Guide*.

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

You can change the following multicast parameters on a global basi:.

- Maximum number of PIM or DVMRP groups – You can change the maximum number of PIM or DVMRP groups for which the software will allocate memory. See “Changing Dynamic Memory Allocation for IP Multicast Groups” on page 14-2.
- Internet Group Membership Protocol (IGMP) V1 and V2 parameters – You can change IGMP parameters globally, such as query interval, group membership time, and maximum response time. You can also add interfaces to IGMP multicast groups. See “Changing IGMP V1 and V2 Parameters” on page 14-4.
- 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. See the following sections:
  - “Enabling Hardware Forwarding of Multicast Traffic On Tagged Ports (JetCore only)” on page 14-6
  - “Enabling Hardware Forwarding for all Fragments of IP Multicast Packets” on page 14-9
  - “JetCore Hardware Forwarding of Multicast Traffic on Tagged and Untagged Ports” on page 14-9
- Designated router election priority for PIM v2 – In PIM v2, you can assign a designated router election priority to each router participating in the multicast. The router with the highest DR election priority is elected the DR. See “Specifying a Designated Router Election Priority for PIM V2” on page 14-12.

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: Beginning with software release 07.6.02, 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 Membership

To increase the number of IGMP membership interfaces you can have for PIM, enter commands such as the following:

```
BigIron(config)# system-max pim-max-int-group 4000
BigIron(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 beginning with software release 07.6.02 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:

```
BigIron(config)# system-max dvmrp-max-int-group 3000
BigIron(config)# write memory
```

NOTE: The **system-max dvmrp-max-int-group** command is no longer available beginning with software release 07.6.02 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 to place these changes into effect.

Defining the Maximum Number of Multicast Flows

The Multicast Flow table is shared by PIM and DVMRP. It defines the maximum number of flows for a PIM or DVMRP multicast switching that can be written in hardware (CAM). To define the maximum number of entries for the Multicast Flow table, enter a command such as the following:

```
BigIron(config)# system-max multicast-flow 2048
```

Syntax: system-max multicast-flow <num>

The <num> parameter specifies the maximum number of PIM and DVMRP multicast cache flows that can be stored in the CAM. Enter a number from 512 – 2048. The default is 1024.

NOTE: Do not set this maximum too high since you may run out of resources in the CAM.

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:

```
BigIron(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:

```
BigIron(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:

```
BigIron(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.

USING THE CLI

To modify the default value for the IGMP (V1 and V2) query interval, enter the following:

```
BigIron(config)# ip igmp query 120
```

Syntax: ip igmp query-interval <1-3600>

USING THE WEB MANAGEMENT INTERFACE

To modify the default value for the IGMP query interval:

1. Log on to the device using a valid user name and password for read-write access. The System configuration panel is displayed.
2. Click on the plus sign next to Configure in the tree view to display the configuration options.
3. Click on the plus sign next to DVMRP in the tree view to display the DVMRP configuration options.
4. Select the IGMP link to display the IGMP configuration panel.
5. Enter a value from 1 – 3600 in the Query Interval field.
6. Click the Apply button to save the change to the device's running-config file.
7. 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.

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.

USING THE CLI

To define an IGMP (V1 and V2) membership time of 240 seconds, enter the following:

```
BigIron(config)# ip igmp group-membership-time 240
```

Syntax: ip igmp group-membership-time <1-7200>

USING THE WEB MANAGEMENT INTERFACE

To modify the default value for the IGMP (V1 and V2) membership time, you would do the following:

1. Log on to the device using a valid user name and password for read-write access. The System configuration panel is displayed.
2. Click on the plus sign next to Configure in the tree view to display the configuration options.
3. Click on the plus sign next to DVMRP in the tree view to display the DVMRP configuration options.
4. Select the IGMP link to display the IGMP configuration panel.
5. Enter a value from 1 – 7200 in the Group Membership Time field.
6. Click the Apply button to save the change to the device's running-config file.
7. 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.

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.

USING THE CLI

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:

```
BigIron(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.

USING THE WEB MANAGEMENT INTERFACE

You cannot change this parameter using the Web management interface.

Adding an Interface to a Multicast Group

You can manually add an interface to an IGMP 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:

```
BigIron(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:

```
BigIron(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**

Enabling Hardware Forwarding of Multicast Traffic On Tagged Ports (JetCore only)

Software release 07.5.06 and later supports IPC and IGC versions that can forward multicast traffic on tagged ports in hardware instead of sending the traffic to the CPU for forwarding. When you use these versions, multicast traffic that needs to be forwarded on a tagged port is forwarded in hardware.

NOTE: This enhancement applies to Layer 3 multicast traffic on JetCore Layer 3 Switches only. All Layer 2 multicast traffic on JetCore or IronCore devices is forwarded by the CPU.

Previous releases and the current release already provide hardware forwarding of multicast traffic on untagged ports.

- If all multicast traffic is on untagged ports, the traffic is forwarded in hardware on JetCore or IronCore.
- If any of the ports forwarding multicast traffic is a tagged port, an IronCore device forwards all the multicast traffic in software. A JetCore device forwards the traffic in hardware if the device contains the required IGC and IPC versions and the support is enabled.

IPC and IGC Requirements

Multicast hardware forwarding for tagged ports requires at least the following IPC and IGC versions:

- IPC version 300 (ASIC version 0x48) or later
- IGC version 400 (ASIC version 0x49) or later

To determine the IPC and IGC versions in your device, enter the **show version** command. Here is an example. The version information is shown in bold type. In this example, the IPCs on the module in slot 13 are the required version but the IGCs on the module in slot 1 are earlier than the required version.

```
BigIron# show version
  SW: Version 07.5.06b1T53 Copyright (c) 1996-2002 Foundry Networks.
      Compiled on Sep 05 2002 at 16:00:44 labeled as B2R07506b1
      (3849519 bytes) from Secondary 07506r.bin
  HW: BigIron 15000 Router, SYSIF version 21
==========================================================================
SL 1: J-BxGMR4 JetCore Management Module, SYSIF 2 (Mini GBIC), M4, ACTIVE
      Serial #:   PR15021840
 4096 KB BRAM, JetCore ASIC IGC version 47, BIA version 88
32768 KB PRAM and 2M-Bit*1 CAM for IGC  0, version 0447
32768 KB PRAM and 2M-Bit*1 CAM for IGC  1, version 0447
==========================================================================
SL 13: J-B48E JetCore Copper E Module, SYSIF 2
      Serial #:   SA26020347
 4096 KB BRAM, JetCore ASIC IPC version 48, BIA version 89
 8192 KB PRAM and 2M-Bit*1 CAM for IPC 48, version 1848
 8192 KB PRAM and 2M-Bit*1 CAM for IPC 49, version 1848
==========================================================================
Active management module:
  466 MHz Power PC processor 750 (version 8/8302) 65 MHz bus

  512 KB boot flash memory
16384 KB code flash memory
  256 KB SRAM
  512 MB DRAM
The system uptime is 3 minutes 42 seconds
The system : started=warm start   reloaded=by "reload"
==========================================================
```

NOTE: All IPCs and IGCs on the device must have at least the versions listed above. Otherwise, the hardware forwarding is disabled and the device uses the CPU to forward multicast traffic.

Disabling or Re-Enabling Hardware Multicast Forwarding For Tagged Ports

If PIM or DVMRP is enabled on the device, and all the IPCs and IGCs in the device are at least the versions listed in “IPC and IGC Requirements” , hardware multicast forwarding for tagged ports is enabled by default. However, if the device will be forwarding multicast traffic on untagged ports only or will not be forwarding any multicast traffic at all, Foundry recommends that you disable the hardware multicast forwarding for tagged ports. The feature uses CAM resources even if none of the tagged ports are actually forwarding multicast traffic.

NOTE: JetCore and IronCore devices still forward all multicast traffic on untagged ports in hardware, regardless of whether multicast hardware forwarding for tagged ports is enabled or disabled.

To disable hardware multicast forwarding for tagged ports, enter the following command at the global CONFIG level of the CLI:

```
BigIron(config)# mcast-hw-replic-disable
BigIron(config)# write memory
BigIron(config)# end
BigIron# reload
```

Syntax: [no] mcast-hw-replic-disable

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

The feature is enabled by default if PIM or DVMRP is enabled and all IPCs and IGCs are the required versions. You also can manually re-enable the feature by entering the following command:

```
BigIron(config)# no mcast-hw-replic-disable
```

NOTE: If the device does not contain the required version for all IPCs and IGCs, entering the command to enable the feature does not result in the feature being enabled.

Enabling Hardware Forwarding of Multicast Traffic in One-Armed-Router Configurations

As described above, JetCore devices running software releases later than 07.5.05D forward multicast traffic destined to multiple VLANs on tagged ports, without the need to send the traffic to the CPU for forwarding. This support is described in “Enabling Hardware Forwarding of Multicast Traffic On Tagged Ports (JetCore only)” on page 14-6. However, this default behavior does not apply to one-armed-router configurations, in which traffic received on a port is destined to another VLAN on the same port.

For example, assume that ports 1/1 and 1/2 are members of two port-based VLANs (10 and 20), and each VLAN has a virtual routing interface. If port 1/1 receives multicast traffic from VLAN 10 and needs to forward the traffic to the virtual routing interface on VLAN 20, the device forwards the traffic to port 1/2 in hardware but uses the CPU to process the same traffic for forwarding back onto port 1/1.

You can enable the device to forward multicast traffic in hardware even in one-armed-router configurations. When you enable this support, the devices still forward multicast traffic between ports in hardware.

NOTE: You cannot use sFlow or port monitoring and hardware forwarding of multicast traffic in one-armed-router configurations on the same device. If you plan to enable hardware forwarding of multicast traffic in one-armed-router configurations, you must first make sure that sFlow and port monitoring are disabled on all ports. If either of these features is enabled when you enable multicast traffic in one-armed-router configurations, you may get unexpected results.

To enable hardware forwarding of multicast traffic in one-armed-router configurations, enter the following commands:

```
BigIron(config)# mcast-hw-replic-oar
BigIron(config)# write memory
BigIron(config)# end
BigIron# reload
```

Syntax: [no] mcast-hw-replic-oar

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

NOTE: For hardware forwarding of multicast traffic in one-armed-router configurations to take effect, hardware forwarding of multicast traffic on tagged ports must be enabled. If you disable hardware forwarding of multicast traffic on tagged ports (by entering the **mcast-hw-replic-disable** command), the **mcast-hw-replic-oar** command does not take effect.

Displaying the State of Hardware Multicast Forwarding

To determine whether hardware multicast forwarding is enabled, enter either of the following commands:

- **show ip pim resource**
- **show ip dvmrp resource**

The last line of the display shows the state of hardware multicast forwarding. Here is an example.

```
BigIron# show ip pim resource
                allocated    in-use available alloc-fail upper-limit
flow                 1022         0      1022          0
PIM mcache           1024         0      1023          0
NBR list               64         0        64          0  64
interface group       256         0       256          0  2048
global group          256         0       256          0  1024
timer                 256         0       256          0  1024
prune nbr            1024         0      1024          0  4096
prune                 128         0       128          0  256
join/prune elem     12240         0     12240          0  48960
pimsm OIF             256         0       256          0  no-limit
IGMP group            256         0       256          0  1024
HW tagged replication enabled
```

Enabling Hardware Forwarding for all Fragments of IP Multicast Packets

NOTE: For JetCore devices running software release 07.6.02 or later, refer to the section "JetCore Hardware Forwarding of Multicast Traffic on Tagged and Untagged Ports" on page 14-9 for details about configuring hardware forwarding of multicast traffic.

By default, a Foundry Layer 3 Switch forwards the first fragment of a fragmented IP multicast packet through hardware, but forwards the remaining fragments through the software. You can enable the device to forward all the fragments of fragmented IP multicast packet through hardware.

NOTE: This feature applies only to Layer 3 Switches, not to Layer 2 Switches.

To enable hardware forwarding of all the IP multicast fragments, use the following CLI method.

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

USING THE CLI

To enable hardware forwarding of all IP multicast fragments, enter the following commands:

```
BigIron(config)# ip multicast-perf
BigIron(config)# write memory
BigIron(config)# end
BigIron# reload
```

Syntax: [no] ip multicast-perf

USING THE WEB MANAGEMENT INTERFACE

You cannot configure this feature using the Web management interface.

JetCore Hardware Forwarding of Multicast Traffic on Tagged and Untagged Ports

Software release 07.6.02 adds support for JetCore devices to perform hardware forwarding of Layer 2 multicast forwarding on tagged and untagged ports. Previous releases sent this traffic to the CPU for forwarding (that is, forwarded it in software).

In release 07.5.05p, support was also added for hardware forwarding of Layer 3 multicast traffic on tagged ports. Since support had already existed for hardware forwarding of Layer 3 multicast traffic on untagged ports in releases prior to 07.5.06, this means that starting with release 07.6.02, JetCore devices support hardware forwarding of multicast Layer 2 and Layer 3 traffic on both tagged and untagged ports.

The following table summarizes these enhancements.

Table 14.1: Multicast forwarding on JetCore devices

	Layer 2 Traffic	Layer 3 Traffic
Tagged	Hardware, starting in Release 07.6.02	Hardware, starting in Release 07.5.05p
Untagged	Hardware, starting in Release 07.6.02	Hardware

Hardware forwarding for multicast traffic on JetCore devices is automatically enabled if the following requirements are met:

- PIM or DVMRP is enabled on the JetCore device.
- The hardware multicast replication feature has not been disabled on the JetCore device. See “Disabling or Re-Enabling Hardware Multicast Forwarding” on page 14-10.

NOTE: If you plan to use hardware forwarding for multicast traffic on a JetCore device, contact your Foundry account representative for additional requirements that may apply to your installation.

When hardware forwarding of multicast traffic is enabled, multicast traffic may still be forwarded in software if one of the following occurs:

- Registration packets in PIM Sparse mode are sent or received.
- Packets are coming from or going to a tunnel. (PIM Dense mode and DVMRP support tunnels.)
- The PIM flow multicast cache is not available. The PIM flow multicast cache is created after two packets for the same group and source address are received by the hardware. You can check if a PIM flow multicast cache is available by using the **show ip pim flow** command.
- The one-armed-router feature is being used, but this feature is not enabled.

NOTE: The time-to-live value in the IP header of a Layer 3 routed packet is not decremented if the packet is also Layer 2-switched in the JetCore hardware.

Disabling or Re-Enabling Hardware Multicast Forwarding

If PIM or DVMRP is enabled on the device, hardware multicast forwarding is enabled by default. However, if the device will be forwarding multicast traffic on untagged ports only or will not be forwarding any multicast traffic at all, Foundry recommends that you disable the hardware multicast forwarding for tagged ports. The feature uses CAM resources even if none of the tagged ports are actually forwarding multicast traffic.

NOTE: JetCore and IronCore devices still forward all multicast traffic on untagged ports in hardware, regardless of whether multicast hardware forwarding for tagged ports is enabled or disabled.

To disable hardware multicast forwarding for tagged ports, enter the following command at the global CONFIG level of the CLI:

```
BigIron(config)# mcast-hw-replic-disable
```

Syntax: [no] mcast-hw-replic-disable

The feature is enabled by default if PIM or DVMRP is enabled on the JetCore device. You also can manually re-enable the feature by entering the following command:

```
BigIron(config)# no mcast-hw-replic-disable
```


Enabling Hardware Forwarding of Multicast Traffic in One-Armed-Router Configurations

JetCore devices running software release 07.5.05p or later forward multicast traffic destined to multiple VLANs on tagged ports in hardware by default, without the need to send the traffic to the CPU for forwarding. However, this default behavior does not apply to one-armed-router configurations, in which traffic received on a port is destined to another VLAN on the same port.

For example, assume that ports 1/1 and 1/2 are members of two port-based VLANs (10 and 20), and each VLAN has a virtual routing interface. If port 1/1 receives multicast traffic from VLAN 10 and needs to forward the traffic to the virtual routing interface on VLAN 20, the device forwards the traffic to port 1/2 in hardware but uses the CPU to process the same traffic for forwarding back onto port 1/1.

You can enable the device to forward multicast traffic in hardware even in one-armed-router configurations. When you enable this support, the devices still forward multicast traffic between ports in hardware.

NOTE: You cannot use sFlow or port monitoring and hardware forwarding of multicast traffic in one-armed-router configurations on the same device. If you plan to enable hardware forwarding of multicast traffic in one-armed-router configurations, you must first make sure that sFlow and port monitoring are disabled on all ports. If either of these features is enabled when you enable multicast traffic in one-armed-router configurations, you may get unexpected results.

To enable hardware forwarding of multicast traffic in one-armed-router configurations, enter the following commands:

```
BigIron(config)# mcast-hw-replic-oar
BigIron(config)# write memory
BigIron(config)# end
BigIron# reload
```

Syntax: [no] mcast-hw-replic-oar

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

NOTE: For hardware forwarding of multicast traffic in one-armed-router configurations to take effect, hardware forwarding of multicast traffic on tagged ports must be enabled. If you disable hardware forwarding of multicast traffic on tagged ports (by entering the **mcast-hw-replic-disable** command), the **mcast-hw-replic-oar** command does not take effect.

Displaying the State of Hardware Multicast Forwarding

To determine whether hardware multicast forwarding is enabled, enter either of the following commands:

- **show ip pim resource**
- **show ip dvmrp resource**

The last line of the display shows the state of hardware multicast forwarding. Here is an example.

```
BigIron# show ip pim resource
                allocated    in-use available alloc-fail upper-limit
flow                 1022         0      1022         0
PIM mcache           1024         0      1023         0
NBR list               64         0        64         0  64
interface group       256         0       256         0  2048
global group          256         0       256         0  1024
timer                 256         0       256         0  1024
prune nbr            1024         0      1024         0  4096
prune                 128         0       128         0  256
join/prune elem     12240         0     12240         0  48960
pimsm OIF             256         0       256         0  no-limit
IGMP group            256         0       256         0  1024
HW tagged replication enabled
```

Specifying a Designated Router Election Priority for PIM V2

In a multi-access network, where two or more routers' interfaces are connected together, it is possible that a packet could reach more than one router. In this kind of network, PIM elects a designated router (DR), whose duties include sending registration or join/prune messages for hosts and forwarding traffic for the LAN segment.

In releases prior to 08.0.00, the router with the highest IP address was elected the DR. Starting in release 08.0.00, you can optionally assign each router a DR election priority. The router with the highest DR election priority is elected the DR. If two or more routers all have the same DR election priority, then the router with the highest IP address is elected the DR. If any of the routers in the multi-access network do not support the DR election priority feature, then the router with the highest IP address is elected the DR.

The DR election priority is set on a per-interface basis. For example, to set a DR election priority of 99 for VE 20, enter the following commands:

```
BigIron(config)# int ve 20
BigIron(config-vif-20)# ip pim dr-priority 99
BigIron(config-vif-20)# exit
```

Syntax: [no] ip pim dr-priority <num>

The priority specified by <num> can be from 0 – 4294967295. The default is 1. Starting in this release, the PIM hello message sent by Foundry devices always contains the router's DR election priority (either the specified priority or 1 if no priority was specified).

To prevent an interface from being elected a DR, you can either set its priority to 0, or you can increase the priority of all interfaces of other routers in the multi-access network to a number greater than 1.

The output of the **show ip pim interface** command has been enhanced to display the device's DR election priority. For example:

```
BigIron# show ip pim interface
Interface v20
PIM Sparse
TTL Thres: 1, Enabled, dr-priority=99, DR: itself
Local Address: 1.1.20.4
Neighbor:
  1.1.20.5
```


PIM Dense

NOTE: This section describes the “dense” mode of PIM, described in RFC 1075. See “PIM Sparse” on page 14-24 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 14.1. 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 14.1, 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 14.1 the sender with address 207.95.5.1 is sending multicast packets to the group 229.225.0.1. If a PIM router receives any groups other than that group, the router discards the group and sends a prune message to the upstream PIM router.

In Figure 14.2, Router R5 is a leaf node with no group members in its IGMP database. Therefore, the router must be pruned from the multicast tree. R5 sends a prune message upstream to its neighbor router R4 to remove itself from the multicast delivery tree and install a prune state, as seen in Figure 14.2. Router 5 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 R4, if both R5 and R6 are in a prune state at the same time, R4 becomes a leaf node with no downstream interfaces and sends a prune message to R1. With R4 in a prune state, the resulting multicast delivery tree would consist only of leaf nodes R2 and R3.

Figure 14.1 Transmission of multicast packets from the source to host group members

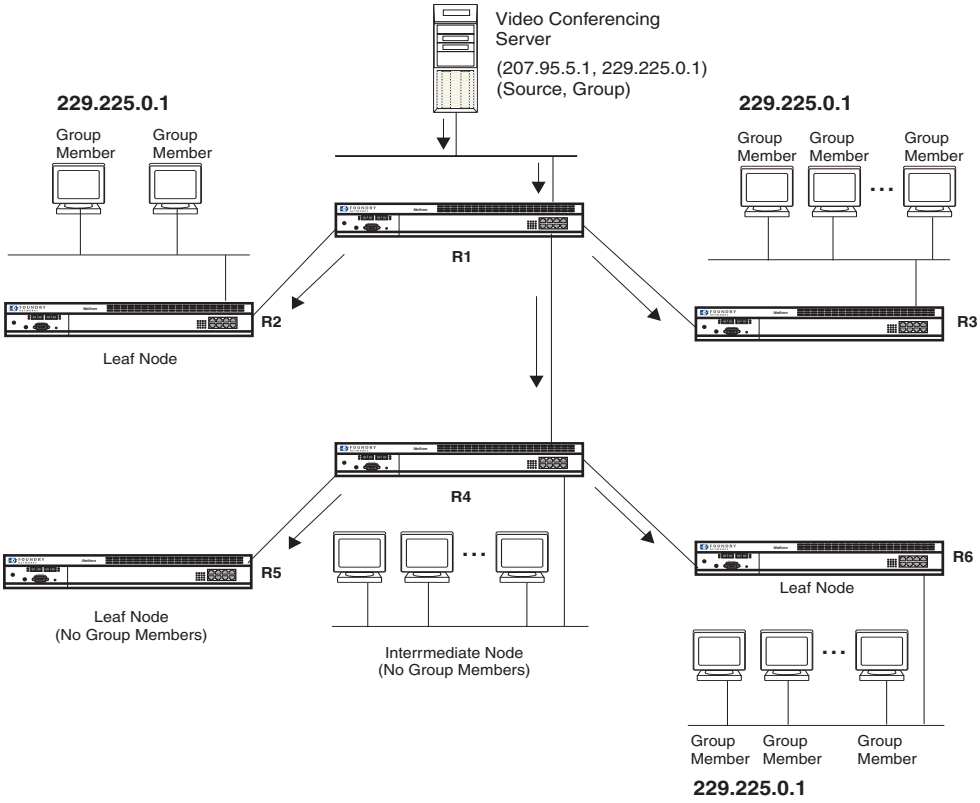
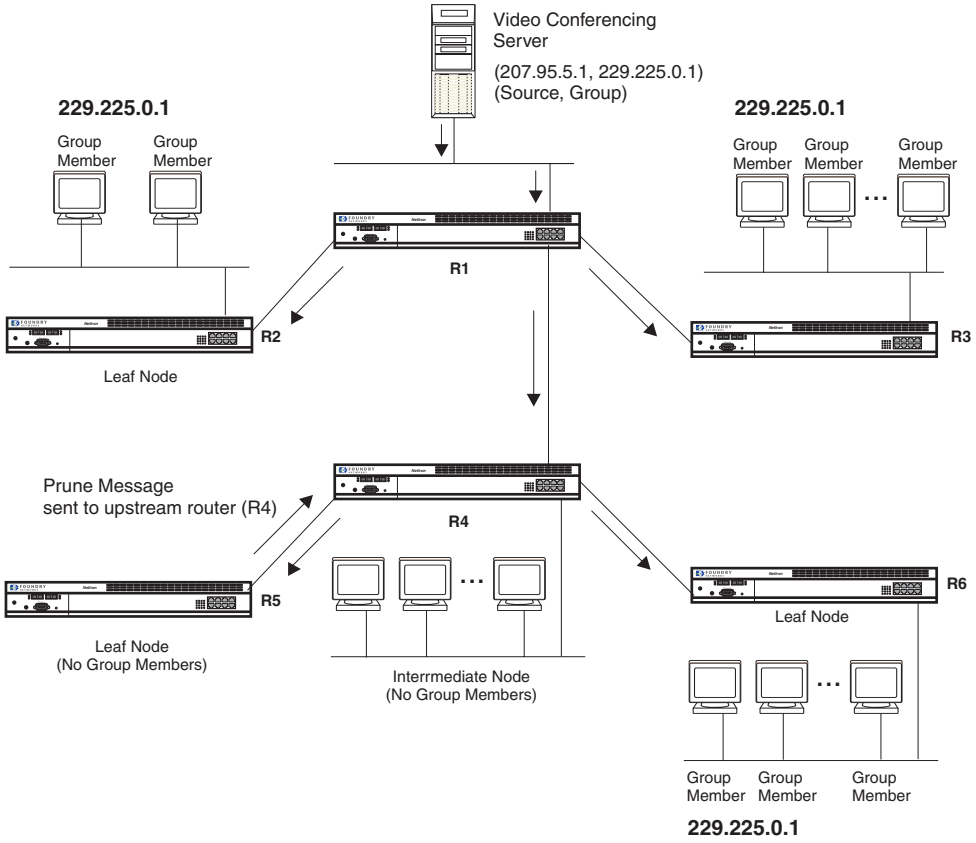


Figure 14.2 Pruning leaf nodes from a multicast tree



Grafts to a Multicast Tree

A PIM router restores pruned branches to a multicast tree by sending graft messages towards the upstream router. Graft messages start at the leaf node and travel up the tree, first sending the message to its neighbor upstream router.

In the example above, if a new 229.255.0.1 group member joins on router R6, which was previously pruned, a graft is sent upstream to R4. Since the forwarding state for this entry is in a prune state, R4 sends a graft to R1. Once R4 has joined the tree, R4 along with R6 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

Software release 07.2.05 and higher supports PIM DM V1 and V2. Previous versions support V1 only. The default in previous releases is V1. The default in release 07.2.05 and higher 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 in software release 07.2.05 and higher. Previous releases support only version 1. 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 software release 07.2.05 or higher and also 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 14-25 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 14.1 on page 14-14.

PIM is enabled on each of the Foundry routers shown in Figure 14.1, 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:

```
BigIron(config)# router pim
```

Syntax: [no] router pim

NOTE: When PIM routing is enabled on a JetCore device, the line rate for receive traffic is reduced by about 5%. The reduction occurs due to overhead from the VLAN multicasting feature, which PIM routing uses. This behavior is normal and does not indicate a problem with the device.

In software releases prior to software release 07.8.00, the behavior of the **[no] router pim** command was as follows:

- Foundry Layer 3 Switches required a software reload whenever you enabled PIM using the **router pim** command.
- Entering a **no router pim** command removed all configuration for PIM multicast on a Layer 3 Switch (**router pim** level) and all PIM and PIM-Sparse (**ip pim** and **ip pim-sparse)** configuration on all interfaces.

Beginning with software release 0 7.8.00:

- 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 PIM configuration. If you want to disable PIM without deleting any PIM configuration, enter the following command:

```
BigIron(config)# router pim
BigIron(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:

```
BigIron(config)# router pim
BigIron(config)# int e 3
BigIron(config-if-3)# ip address 207.95.5.1/24
BigIron(config-if-3)# ip pim
BigIron(config-if-3)# write memory
BigIron(config-if-3)# end
BigIron# 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:

```
BigIron(config-if-1/1)# ip pim version 2
```

```
BigIron(config-if-1/1)# no ip pim version 1
```

To disable PIM DM on the interface, enter the following command:

```
BigIron(config-if-1/1)# no ip pim
```

USING THE WEB MANAGEMENT INTERFACE

1. Log on to the device using a valid user name and password for read-write access.
2. If you have not already enabled PIM, enable it by clicking on the Enable radio button next to PIM on the System configuration panel, then clicking Apply to apply the change.
3. Click on the plus sign next to Configure in the tree view to expand the list of configuration options.
4. Click on the plus sign next to PIM in the tree view to expand the list of PIM option links.
5. Click on the Virtual Interface link to display the PIM Interface configuration panel.

NOTE: If the device already has PIM interfaces, a table listing the interfaces is displayed. Click the Modify button to the right of the row describing an interface to change its configuration, or click the Add Virtual Interface link to display the PIM Interface configuration panel.

---

6. Select the interface type. You can select Subnet or Tunnel.
7. Select the IP address of the interface being configured from the Local Address pulldown menu.
8. If you are configuring an IP Tunnel, enter the IP address of the destination interface, the end point of the IP Tunnel, in the Remote Address field. IP tunneling must also be enabled and defined on the destination router interface as well.

NOTE: The Remote Address field applies only to tunnel interfaces, not to subnet interfaces.

9. Modify the time to live threshold (TTL) if necessary. The TTL defines the minimum value required in a packet in order for the packet to be forwarded out the interface.

NOTE: 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 values are 1 – 64. The default value is 1.

10. Click the Add button to save the change to the device's running-config file.
11. 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.
12. Click on the plus sign next to Command in the tree view to list the command options.
13. Select the Reload link and select Yes when prompted to reload the software. You must reload after enabling PIM to place the change into effect. If PIM was already enabled when you added the interface, you do not need to reload.

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.

USING THE CLI

To apply a PIM neighbor timeout value of 360 seconds to all ports on the router operating with PIM, enter the following:

```
BigIron(config)# router pim
BigIron(config-pim-router)# nbr-timeout 360
```

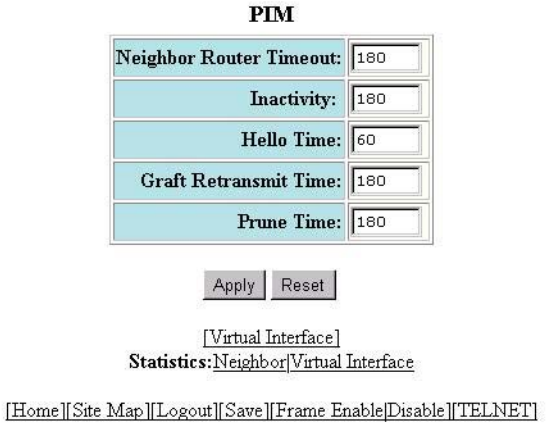
Syntax: nbr-timeout <60-8000>

The default is 180 seconds.

USING THE WEB MANAGEMENT INTERFACE

1. Log on to the device using a valid user name and password for read-write access.
2. Click on the plus sign next to Configure in the tree view to expand the list of configuration options.
3. Click on the plus sign next to PIM in the tree view to expand the list of PIM option links.

4. Click on the General link to display the PIM configuration panel, as shown in the following example.



5. Enter a value from 10 – 3600 into the Neighbor Router Timeout field.
6. Click the Apply button to save the change to the device's running-config file.
7. 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.

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.

USING THE CLI

To apply a PIM hello timer of 120 seconds to all ports on the router operating with PIM, enter the following:

```
BigIron(config)# router pim
BigIron(config-pim-router)# hello-timer 120
```

Syntax: hello-timer <10-3600>

The default is 60 seconds.

USING THE WEB MANAGEMENT INTERFACE

1. Log on to the device using a valid user name and password for read-write access.
2. If you have not already enabled PIM, enable it by clicking on the Enable radio button next to PIM on the System configuration panel, then clicking Apply to apply the change.
3. Click on the plus sign next to Configure in the tree view to expand the list of configuration options.
4. Click on the plus sign next to PIM in the tree view to expand the list of PIM option links.
5. Click on the General link to display the PIM configuration panel
6. Enter a value from 10 – 3600 into the Prune Time field.
7. Click the Apply button to save the change to the device's running-config file.
8. 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.

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.

USING THE CLI

To set the PIM prune timer to 90, enter the following:

```
BigIron(config)# router pim
BigIron(config-pim-router)# prune-timer 90
```

Syntax: prune-timer <10-3600>

The default is 180 seconds.

USING THE WEB MANAGEMENT INTERFACE

1. Log on to the device using a valid user name and password for read-write access.
2. If you have not already enabled PIM, enable it by clicking on the Enable radio button next to PIM on the System configuration panel, then clicking Apply to apply the change.
3. Click on the plus sign next to Configure in the tree view to expand the list of configuration options.
4. Click on the plus sign next to PIM in the tree view to expand the list of PIM option links.
5. Click on the General link to display the PIM configuration panel
6. Enter a value from 10 – 3600 in the Hello Time field.
7. Click the Apply button to save the change to the device's running-config file.
8. 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.

Modifying the Prune Wait Timer

Beginning with software release 07.6.04, a new 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:

```
BigIron(config)# router pim
BigIron(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 following command at any level of the CLI:

```
BigIron(config)#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.

USING THE CLI

To change the graft retransmit timer from the default of 180 to 90 seconds, enter the following:

```
BigIron(config)# router pim
BigIron(config-pim-router)# graft-retransmit-timer 90
```

Syntax: graft-retransmit-timer <10-3600>

The default is 180 seconds.

USING THE WEB MANAGEMENT INTERFACE

1. Log on to the device using a valid user name and password for read-write access.
2. If you have not already enabled PIM, enable it by clicking on the Enable radio button next to PIM on the System configuration panel, then clicking Apply to apply the change.
3. Click on the plus sign next to Configure in the tree view to expand the list of configuration options.
4. Click on the plus sign next to PIM in the tree view to expand the list of PIM option links.
5. Click on the General link to display the PIM configuration panel
6. Enter a value from 10 – 3600 into the Graft Retransmit Time field.
7. Click the Apply button to save the change to the device's running-config file.
8. 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.

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.

USING THE CLI

To apply a PIM inactivity timer of 90 seconds to all PIM interfaces, enter the following:

```
BigIron(config)# router pim
BigIron(config-pim-router)# inactivity-timer 90
```

Syntax: inactivity-timer <10-3600>

The default is 180 seconds.

USING THE WEB MANAGEMENT INTERFACE

1. Log on to the device using a valid user name and password for read-write access.
2. If you have not already enabled PIM, enable it by clicking on the Enable radio button next to PIM on the System configuration panel, then clicking Apply to apply the change.
3. Click on the plus sign next to Configure in the tree view to expand the list of configuration options.
4. Click on the plus sign next to PIM in the tree view to expand the list of PIM option links.
5. Click on the General link to display the PIM configuration panel
6. Enter a value from 10 – 3600 into the Inactivity field.
7. Click the Apply button to save the change to the device's running-config file.
8. 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.

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
```

The Highest IP RPF feature was introduced in Enterprise software release 07.6.06. When this 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:

```
BigIron(config)# router pim
BigIron(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.

USING THE WEB MANAGEMENT INTERFACE

You cannot configure this feature using the Web management interface.

Failover Time in a Multi-Path Topology

Previously, when a port in a multi-path topology fails, multicast routers could take up to one minute to establish a new path, if the failed port is the input port of the downstream router. Enterprise software release 07.6.06 and later reduces this time. 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 64. The default TTL value is 1.

Configuration Notes

- If the TTL for an interface is greater than 1, PIM packets recieved 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.

USING THE CLI

To configure a TTL of 45, enter the following:

```
BigIron(config-if-3/24)# ip pim ttl 45
```

Syntax: ip pim ttl <1-64>

USING THE WEB MANAGEMENT INTERFACE

To modify the PIM parameter (TTL) for an interface:

1. Log on to the device using a valid user name and password for read-write access.
2. Click on the plus sign next to Configure in the tree view to expand the list of configuration options.
3. Click on the plus sign next to PIM in the tree view to expand the list of PIM option links.
4. Select the Virtual Interface link to display a table listing the configured PIM Interfaces.
5. Click on the Modify button next to the interface you want to modify. The PIM Interface configuration panel is displayed.
6. Modify the parameters as needed.
7. Click the Add button to save the changes to the device's running-config file.
8. 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.

Dropping PIM Traffic in Hardware

Beginning with IronWare software release 07.8.00, 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. Also, this feature applies only to devices with JetCore modules; IronCore modules cannot drop tagged multicast packets in hardware.

When a multicast stream has no output interfaces, the Layer 3 Switch can drop packets in hardware if the multicast traffic meets either of the following conditions:

- The input port of the traffic has no neighbor, so it is not necessary to send a prune message.
- The input port has neighbors and the traffic is Layer 2 with a source IP address that is on the same subnet as the input port. Foundry PIM Dense ignores prune message from a Layer 3 Switch which is on the same subnet as the source.

USING THE CLI

To configure the device to drop PIM traffic in hardware, enter the following command at the **router pim** level:

```
BigIron(config)# router pim
BigIron(config-pim-router)# hardware-drop
```

Syntax: hardware-drop

When you enable the hardware-drop feature, the **show ip pim mcache** command includes “drop” in the flag field if a CAM is installed for the purpose of the drop. For example,

```
BigIron# show ip pim mcache
1   (110.110.110.10 224.1.11.1) in v110 (e4/6), cnt=72
     Source is directly connected
     Sparse Mode, RPT=0 SPT=1 Reg=0
     fast=1 slow=0 pru=0 swL2=0 hwL2=0 drop
     age=60s up-time=183m fid=08ac, cam=3818,
```

USING THE WEB MANAGEMENT INTERFACE

You cannot change this parameter using the Web management interface.

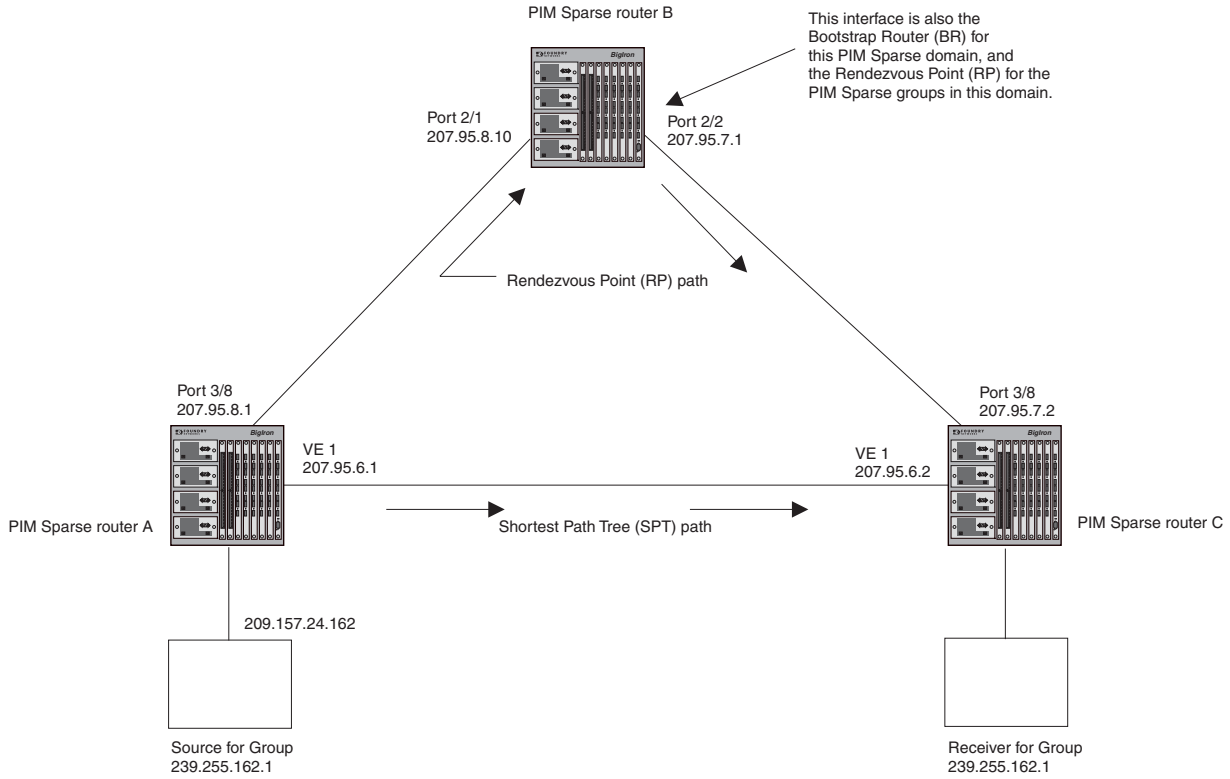
PIM Sparse

Software release 06.5.00 and higher contain support for 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 14.3 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 14.3 Example PIM Sparse domain



PIM Sparse Router Types

Routers that are configured with PIM Sparse interfaces also can be configured to fill one or more of the following roles:

- PMBR – A PIM router 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 routers within the domain. Each PIM Sparse domain has one active BSR. For redundancy, you can configure ports on multiple routers 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 14.3, PIM Sparse router 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 routers 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 routers. In the example in Figure 14.3, PIM Sparse router 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 14.3 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 router A and the recipient is attached to PIM Sparse router C. PIM Sparse router 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 router A and router C, which bypasses the RP (router 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 routers 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 14.3, Layer 3 Switch A forwards the first packet from group 239.255.162.1's source to the destination by sending the packet to router B, which is the RP. Router B then sends the packet to router C. For the second and all future packets that router A receives from the source for the receiver, router A forwards them directly to router 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 softwsare 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.

NOTE: When PIM routing is enabled on a JetCore device, the line rate for receive traffic is reduced by about 5%. The reduction occurs due to overhead from the VLAN multicasting feature, which PIM routing uses. This behavior is normal and does not indicate a problem with the device.

USING THE CLI

To configure basic global PIM Sparse parameters, enter commands such as the following on each Layer 3 Switch within the PIM Sparse domain:

```
BigIron(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 router without configuring the Layer 3 Switch 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 the Layer 3 Switch as both of these. See “Configuring BSRs” on page 14-27.

In software releases prior to software release 07.8.00, the behavior of the **[no] router pim** command was as follows:

- Foundry Layer 3 Switches required a software reload whenever you enable or disable PIM using the **no router pim** command.
- Entering a **no router pim** command removed all configuration for PIM multicast on a Layer 3 Switch (**router pim** level) and all PIM and PIM-Sparse (**ip pim** and **ip pim-sparse)** configuration on all interfaces.

Beginning with software release 07.8.00:

- 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.

USING THE WEB MANAGEMENT INTERFACE

You cannot configure PIM Sparse parameters using the Web management interface.

Globally Enabling and Disabling PIM without Deleting Multicast Configuration

As stated above entering a **no router pim** command deletes PIM configuration. If you want to disable PIM without deleting any PIM configuration, enter the following command:

```
BigIron(config)# router pim
BigIron(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.

USING THE CLI

To enable PIM Sparse mode on an interface, enter commands such as the following:

```
BigIron(config)# interface ethernet 2/2
BigIron(config-if-2/2)# ip address 207.95.7.1 255.255.255.0
BigIron(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:

```
BigIron(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.

USING THE WEB MANAGEMENT INTERFACE

You cannot configure PIM Sparse parameters using the Web management interface.

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 14-28 for instructions on how to configure RPs.

To configure the Layer 3 Switch as a candidate BSR and RP, use the following CLI method.

USING THE CLI

To configure the Layer 3 Switch as a candidate BSR, enter commands such as the following:

```
BigIron(config)# router pim
BigIron(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 <portnum> | loopback <num> | ve <num> <hash-mask-length> [<priority>]

The **ethernet** <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** <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:

```
BigIron(config-pim-router)# rp-candidate ethernet 2/2
```

Syntax: [no] rp-candidate ethernet <portnum> | loopback <num> | ve <num>

The **ethernet** <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** <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.

```
BigIron(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:

```
BigIron(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.

USING THE WEB MANAGEMENT INTERFACE

You cannot configure PIM Sparse parameters using the Web management interface.

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.

In release 07.6.03, the **clear pim rp-map** command was added to allow 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:

```
BigIron(config)# 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.

USING THE CLI

To specify the IP address of the RP, enter commands such as the following:

```
BigIron(config)# router pim
BigIron(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.

USING THE WEB MANAGEMENT INTERFACE

You cannot configure PIM Sparse parameters using the Web management interface.

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.

USING THE CLI

To change the number of packets the Layer 3 Switch sends using the RP before switching to the SPT, enter commands such as the following:

```
BigIron(config)# router pim
BigIron(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.

USING THE WEB MANAGEMENT INTERFACE

You cannot configure PIM Sparse parameters using the Web management interface.

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.

USING THE CLI

To change the Join/Prune interval, enter commands such as the following:

```
BigIron(config)# router pim
BigIron(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.

USING THE WEB MANAGEMENT INTERFACE

You cannot configure PIM Sparse parameters using the Web management interface.

Specifying a Designated Router Election Priority for PIM V2

In a multi-access network, where two or more routers' interfaces are connected together, it is possible that a packet could reach more than one router. In this kind of network, PIM elects a designated router (DR), whose duties include sending registration or join/prune messages for hosts and forwarding traffic for the LAN segment. The router with the highest IP address was elected the DR.

However, on devices running Enterprise software release 08.0.00, you can optionally assign each router a DR election priority. The router with the highest DR election priority is elected the DR. If two or more routers all have the same DR election priority, then the router with the highest IP address is elected the DR. If any of the routers in the multi-access network do not support the DR election priority feature, then the router with the highest IP address is elected the DR.

The DR election priority is set on a per-interface basis. For example, to set a DR election priority of 99 for VE 20, enter the following commands:

```
BigIron(config)# int ve 20
BigIron(config-vif-20)# ip pim dr-priority 99
BigIron(config-vif-20)# exit
```

Syntax: [no] ip pim dr-priority <num>

The priority specified by <num> can be from 0 – 4294967295. The default is 1. Starting in this release, the PIM hello message sent by Foundry devices always contains the router's DR election priority (either the specified priority or 1 if no priority was specified).

To prevent an interface from being elected a DR, you can either set its priority to 0, or you can increase the priority of all interfaces of other routers in the multi-access network to a number greater than 1.

The output of the **show ip pim interface** command has been enhanced to display the device's DR election priority. For example:

```
BigIron# show ip pim interface
Interface v20
PIM Sparse
TTL Thres: 1, Enabled, dr-priority=99, DR: itself
Local Address: 1.1.20.4
Neighbor:
  1.1.20.5
```

Dropping PIM Traffic in Hardware

Beginning with IronWare software release 07.8.00, 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 "Dropping PIM Traffic in Hardware" on page 14-23.

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, use the following CLI method.

USING THE CLI

To display PIM Sparse configuration information, enter the following command at any CLI level:

```
BigIron(config-pim-router)# 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 14.3.

This display shows the following information.

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.
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 14-30.
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.

This Field...	Displays...
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.

USING THE WEB MANAGEMENT INTERFACE

You cannot display PIM Sparse information using the Web management interface.

Displaying a List of Multicast Groups

To display a list of the IP multicast groups the Layer 3 Switch is forwarding, use the following CLI method.

USING THE CLI

To display PIM Sparse configuration information, enter the following command at any CLI level:

```
BigIron(config-pim-router)# 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.

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.

USING THE WEB MANAGEMENT INTERFACE

You cannot display PIM Sparse information using the Web management interface.

Displaying BSR Information

To display information about the BSR, use the following CLI method.

USING THE CLI

To display BSR information, enter the following command at any CLI level:

```
BigIron(config-pim-router)# 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.

```
BigIron(config-pim-router)# 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.

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.

This Field...	Displays...
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.

USING THE WEB MANAGEMENT INTERFACE

You cannot display PIM Sparse information using the Web management interface.

Displaying Candidate RP Information

To display candidate RP information, use the following CLI method.

USING THE CLI

To display candidate RP information, enter the following command at any CLI level:

```
BigIron(config-pim-router)# 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.

```
BigIron(config-pim-router)# show ip pim rp-candidate
```

This system is not a Candidate-RP.

Syntax: show ip pim rp-candidate

This display shows the following information.

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.
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.

USING THE WEB MANAGEMENT INTERFACE

You cannot display PIM Sparse information using the Web management interface.

Displaying RP-to-Group Mappings

To display RP-to-group mappings, use the following CLI method.

USING THE CLI

To display RP-to-group-mappings, enter the following command at any CLI level:

```
BigIron(config-pim-router)# 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.

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.

USING THE WEB MANAGEMENT INTERFACE

You cannot display PIM Sparse information using the Web management interface.

Displaying RP Information for a PIM Sparse Group

To display RP information for a specific PIM Sparse group, use the following CLI method.

USING THE CLI

To display RP information for a PIM Sparse group, enter the following command at any CLI level:

```
BigIron(config-pim-router)# 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.

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.

USING THE WEB MANAGEMENT INTERFACE

You cannot display PIM Sparse information using the Web management interface.

Displaying the RP Set List

To display the RP set list, use the following CLI method.

USING THE CLI

To display the RP set list, enter the following command at any CLI level:

```
BigIron(config)#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.

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.

USING THE WEB MANAGEMENT INTERFACE

You cannot display PIM Sparse information using the Web management interface.

Displaying Multicast Neighbor Information

To display information about the Layer 3 Switch's IP Multicast neighbors, use either of the following methods.

USING THE CLI

To display information about the Layer 3 Switch's PIM neighbors, enter the following command at any CLI level:

```
BigIron(config-pim-router)# 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.

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.

This Field...	Displays...
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.

USING THE WEB MANAGEMENT INTERFACE

1. Log on to the device using a valid user name and password for read-only or read-write access. The System configuration panel is displayed.
2. Click on the plus sign next to Monitor in the tree view.
3. Click on the plus sign next to PIM in the tree view to expand the list of PIM option links.
4. Click on the Neighbor link to display the IP interface table.

Displaying Information About an Upstream Neighbor Device

Beginning in software release 07.7.00, 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:

```
BigIron# show ip pim rpf 1.1.20.2
directly connected or via an L2 neighbor
```

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. For more information about this command, see the *Foundry Switch and Router Command Line Interface Reference*.

The following example outputs show other messages that the Terathon devices BigIron MG8 display with this command.

```
BigIron# show ip pim rpf 1.2.3.4
no route
BigIron# show ip pim rpf 1.10.10.24
upstream neighbor=1.1.20.1 on v21 using ip route
```

Syntax: show ip pim | dvmrp rpf <IP address>

where <IP address> is a valid source IP address

Displaying the PIM Flow Cache

To display the PIM flow cache, use the following CLI method.

USING THE CLI

To display the PIM flow cache, enter the following command at any CLI level:

```
BigIron(config-pim-router)# 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.

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.

USING THE WEB MANAGEMENT INTERFACE

You cannot display the PIM flow cache using the Web management interface.

Displaying the PIM Multicast Cache

To display the PIM multicast cache, use the following CLI method.

USING THE CLI

To display the PIM multicast cache, enter the following command at any CLI level:

```
BigIron(config-pim-router)# 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.

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.

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.

USING THE WEB MANAGEMENT INTERFACE

You cannot display the PIM multicast cache using the Web management interface.

Displaying PIM Traffic Statistics

To display PIM traffic statistics, use the following CLI method.

USING THE CLI

To display PIM traffic statistics, enter the following command at any CLI level:

```
BigIron(config-pim-router)# 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.

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.

USING THE WEB MANAGEMENT INTERFACE

You cannot display PIM statistics using the Web management interface.

Displaying and Clearing PIM Errors

USING THE CLI

If you want to determine how many PIM errors there are on the device, enter the following command:

```
BigIron# 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:

```
BigIron# clear pim counters
```

Syntax: clear pim counters

USING THE WEB MANAGEMENT INTERFACE

You cannot change this parameter using the Web management interface.

Configuring Multicast Source Discovery Protocol (MSDP)

NOTE: This feature is supported on the following devices:

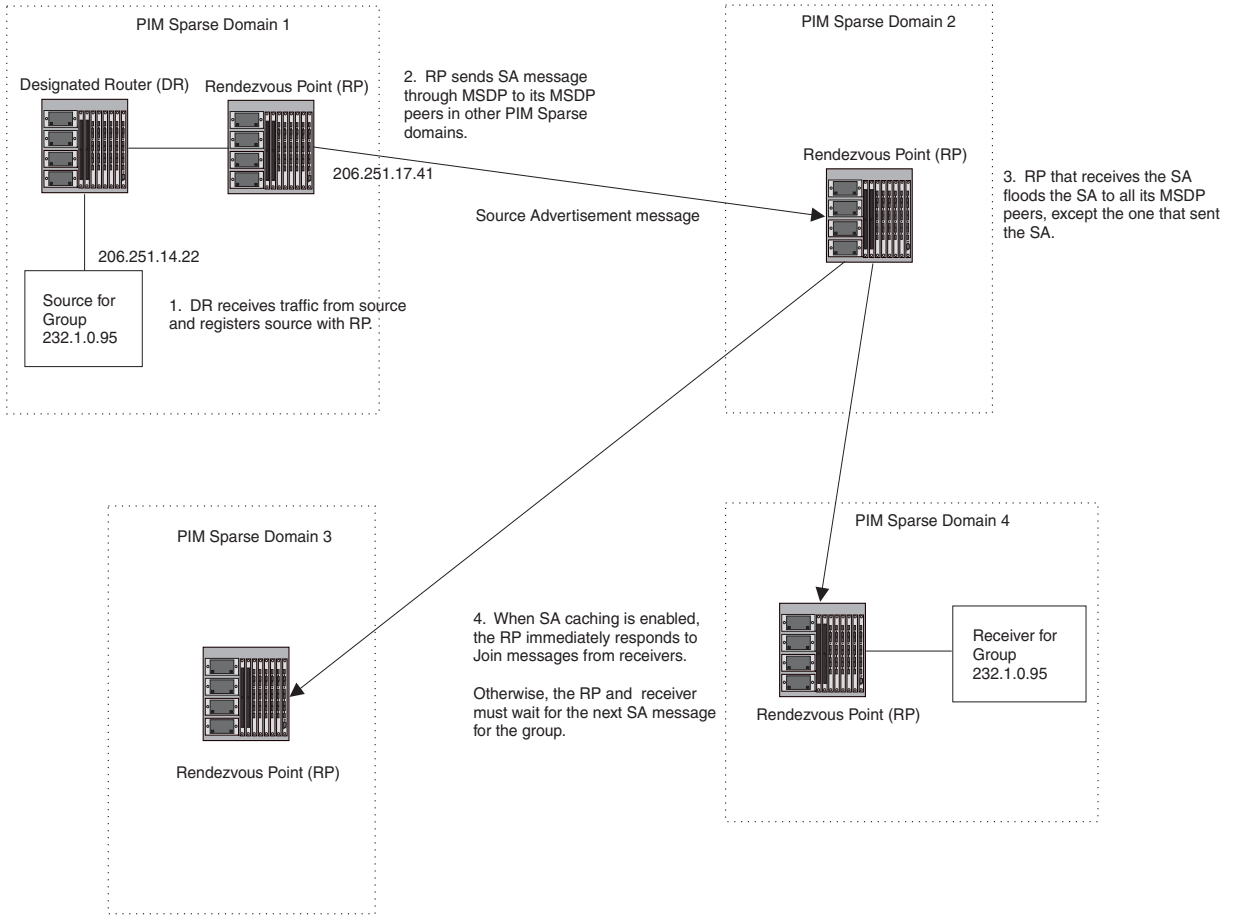
- On devices running Enterprise software release 07.1.00 and later
- On FastIron Edge Switches running software release 01.0.00 and later.
- On BigIron MG8 and NetIron 40G, MSDP is supported in software release 02.2.01 and later.

The Multicast Source Discovery Protocol (MSDP) is used by Protocol Independent Multicast (PIM) Sparse routers to exchange routing information for PIM Sparse multicast groups across PIM Sparse domains. Routers running MSDP can discover PIM Sparse sources that are in other PIM Sparse domains.

PIM Sparse routers use MSDP to register PIM Sparse multicast sources in a domain with the Rendezvous Point (RP) for that domain.

Figure 14.4 shows an example of some PIM Sparse domains. For simplicity, this example show only one Designated Router (DR), one group source, and one receiver for the group. Only one PIM Sparse router within each domain needs to run MSDP.

Figure 14.4 PIM Sparse domains joined by MSDP routers



In this example, the source for PIM Sparse multicast group 232.0.1.95 is in PIM Sparse domain 1. The source sends a packet for the group to its directly attached DR. The DR sends a Group Advertisement message for the group to the domain's RP. The RP is configured for MSDP, which enables the RP to exchange source information with other PIM Sparse domains by communicating with RPs in other domains that are running MSDP.

The RP sends the source information to each of its peers by sending a Source Active message. The message contains the IP address of the source, the group address to which the source is sending, and the IP address of the RP interface with its peer. By default, the IP address included in the RP address field of the SA message is the IP address of the originating RP. Beginning with software release 07.7.00, an SA message can use the IP address of any interface on the originating RP. (The interface is usually a loopback interface.)

In this example, the Source Active message contains the following information:

- Source address: 206.251.14.22
- Group address: 232.1.0.95
- RP address: 206.251.17.41

Figure 14.4 shows only one peer for the MSDP router (which is also the RP here) in domain 1, so the Source Active message goes to only that peer. When an MSDP router has multiple peers, it sends a Source Active message to each of those peers. Each peer sends the Source Advertisement to its other MSDP peers. The RP that receives the Source Active message also sends a Join message for the group if the RP that received the message has receivers for the group.

Peer Reverse Path Forwarding (RPF) Flooding

When the MSDP router (also the RP) in domain 2 receives the Source Active message from its peer in domain 1, the MSDP router in domain 2 forwards the message to all its other peers. The propagation process is sometimes called “peer Reverse Path Forwarding (RPF) flooding”. This term refers to the fact that the MSDP router uses its PIM Sparse RPF tree to send the message to its peers within the tree. In Figure 14.4, the MSDP router floods the Source Active message it receives from its peer in domain 1 to its other peers, in domains 3 and 4.

Note that the MSDP router in domain 2 does not forward the Source Active back to its peer in domain 1, because that is the peer from which the router received the message. An MSDP router never sends a Source Active message back to the peer that sent it. The peer that sent the message is sometimes called the “RPF peer”. The MSDP router uses the unicast routing table for its Exterior Gateway Protocol (EGP) to identify the RPF peer by looking for the route entry that is the next hop toward the source. Often, the EGP protocol is Border Gateway Protocol (BGP) version 4.

NOTE: MSDP depends on BGP and MBGP for interdomain operations.

The MSDP routers in domains 3 and 4 also forward the Source Active message to all their peers except the ones that sent them the message. Figure 14.4 does not show additional peers.

Source Active Caching

When an MSDP router that is also an RP receives a Source Active message, the RP checks its PIM Sparse multicast group table for receivers for the group. If the DR has a receiver for the group being advertised in the Source Active message, the DR sends a Join message for that receiver back to the DR in the domain from which the Source Active message came. Usually, the DR is also the MSDP router that sent the Source Active message.

In Figure 14.4, if the MSDP router and RP in domain 4 has a table entry for the receiver, the RP sends a Join message on behalf of the receiver back through the RPF tree to the RP for the source, in this case the RP in domain 1.

Some MSDP routers that are also RPs can cache Source Active messages. If the RP is not caching Source Active messages, the RP does not send a Join message unless it already has a receiver that wants to join the group. Otherwise, the RP does not send a Join message and does not remember the information in the Source Active message after forwarding it. If the RP receives a request from a receiver for the group, the RP and receiver must wait for the next Source Active message for that group before the RP can send a Join message for the receiver.

However, if Source Active caching is enabled on the MSDP and RP router, the RP caches the Source Active messages it receives. In this case, even if the RP does not have a receiver for a group when the RP receives the Source Active message for the group, the RP can immediately send a Join for a new receiver that wants to join the group, without waiting for the next Source Active message from the RP in the source’s domain.

The size of the cache used to store MSDP Source Active messages is 4 K. On device running Enterprise software release 08.0.00 and later, the size of the cache is 8K.

Configuring MSDP

To configure MSDP on a Layer 3 Switch, perform the following tasks:

- Enable MSDP.
- Configure the MSDP peers.

NOTE: The PIM Sparse Rendezvous Point (RP) is also an MSDP peer.

NOTE: Routers that run MSDP must also run BGP. Also, the source address used by the MSDP router must be the same source address used by BGP.

Enabling MSDP

Use the following CLI method to enable MSDP.

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

USING THE CLI

To enable MSDP, enter the following commands.

```
BigIron(config)# router msdp
BigIron(config-msdp-router)# write memory
BigIron(config-msdp-router)# end
BigIron# reload
```

Syntax: [no] router msdp

USING THE WEB MANAGEMENT INTERFACE

You cannot configure MSDP using the Web management interface.

Configuring MSDP Peers

Use the following CLI method to configure an MSDP peer.

USING THE CLI

To configure an MSDP peer, enter a command such as the following at the MSDP configuration level.

```
BigIron(config-msdp-router)# msdp-peer 205.216.162.1
```

Syntax: [no] msdp-peer <ip-addr> [connect-source loopback <num>]

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

The **connect-source loopback** <num> parameter specifies the loopback interface you want to use as the source for sessions with the neighbor.

NOTE: It is strongly recommended that you use the connect-source loopback <num> parameter when issuing the **msdp-peer** command. If you do not use this parameter, the Layer 3 Switch uses the subnet interface configured on the port.

Also, make sure the IP address of the connect-source loopback is the same as the source IP address used by the MSDP router, the PIM-RP, and the BGP router.

The commands in the following example add an MSDP neighbor and specify a loopback interface as the source interface for sessions with the neighbor. By default, the Layer 3 Switch uses the subnet address configured on the physical interface where you configure the neighbor as the source address for sessions with the neighbor.

```
BigIron(config)# interface loopback 1
BigIron(config-lbif-1)# ip address 9.9.9.9/32
BigIron(config-lbif-1)# interface ethernet 3/1
BigIron(config-if-3/1)# msdp-peer 2.2.2.99 connect-source loopback 1
```

USING THE WEB MANAGEMENT INTERFACE

You cannot configure MSDP using the Web management interface.

Designating an Interface's IP Address as the RP's IP Address

When an RP receives a Source Active message, it checks its PIM Sparse multicast group table for receivers for the group. If it finds a receiver, the RP sends a Join message for that receiver back to the RP that originated the Source Active message. The originator RP is identified by its RP address.

By default, the IP address included in the RP address field of the SA message is the IP address of the originating RP. Beginning with this release, an SA message can use the IP address of any interface on the originating RP. (The interface is usually a loopback interface.)

To designate an interface's IP address to be the IP address of the RP, enter commands such as the following:

```
BigIron(config)# interface loopback 2
BigIron(config-lbif-2)# ip address 2.2.1.99/32
```

```
BigIron(config)# router msdp
BigIron(config-msdp-router)# originator-id loopback 2
BigIron(config-msdp-router)# exit
```

Syntax: [no] originator-id <type> <number>

The **originator-id** parameter instructs MSDP to use the specified address as the IP address of the RP in an SA message. This address must be the address of the interface used to connect the RP to the source. There are no default originator-ids.

The <type> parameter indicates the type of interface used by the RP. Ethernet, loopback and virtual routing interfaces (ve) can be used.

The <number> parameter specifies the interface number (for example: loopback number, port number or virtual routing interface number.)

Filtering MSDP Source-Group Pairs

Software release 07.6.01 and later allows you to filter individual source-group pairs in MSDP Source-Active messages.

- **sa-filter in** – Filters source-group pairs received in Source-Active messages from an MSDP neighbor
- **sa-filter originate** – Filters source-group pairs in Source-Active messages in advertisements to an MSDP neighbor

Filtering Incoming Source-Active Messages

The following example configures filters for incoming Source-Active messages from three MSDP neighbors:

- For peer 2.2.2.99, all source-group pairs in Source-Active messages from the neighbor are filtered out (dropped).
- For peer 2.2.2.97, all source-group pairs except those with 10.x.x.x as the source are permitted.
- For peer 2.2.2.96, all source-group pairs except those associated with RP 2.2.42.3 are permitted.

Example

The following commands configure an IP address on port 3/1. This is the port on which the MSDP neighbors will be configured.

```
BigIron(config)# interface ethernet 3/1
BigIron(config-if-3/1)# ip address 2.2.2.98/24
BigIron(config-if-3/1)# exit
```

The following commands configure a loopback interface. The Layer 3 Switch will use this interface as the source address for communicating with the MSDP neighbors.

```
BigIron(config)# interface loopback 1
BigIron(config-lbif-1)# ip address 9.9.9.8/32
BigIron(config-lbif-1)# exit
```

The following commands configure extended ACLs. The ACLs will be used in route maps, which will be used by the Source-Active filters.

```
BigIron(config)# access-list 123 permit 10.0.0.0 0.255.255.255 any
BigIron(config)# access-list 124 permit 2.2.42.3 0.0.0.0 any
BigIron(config)# access-list 125 permit any any
```


The following commands configure the route maps.

```
BigIron(config)# route-map msdp_map deny 1
BigIron(config-routemap msdp_map)# match ip address 123
BigIron(config-routemap msdp_map)# exit
BigIron(config)# route-map msdp2_map permit 1
BigIron(config-routemap msdp2_map)# match ip address 125
BigIron(config-routemap msdp2_map)# exit
BigIron(config)# route-map msdp2_rp_map deny 1
BigIron(config-routemap msdp2_rp_map)# match ip route-source 124
BigIron(config-routemap msdp2_rp_map)# exit
```

The following commands enable MSDP and configure the MSDP neighbors on port 3/1.

```
BigIron(config)# router msdp
BigIron(config-msdp-router)# exit
BigIron(config)# interface ethernet 3/1
BigIron(config-if-3/1)# msdp-peer 2.2.2.99 connect-source loopback 1
BigIron(config-if-3/1)# msdp-peer 2.2.2.97 connect-source loopback 1
BigIron(config-if-3/1)# msdp-peer 2.2.2.96 connect-source loopback 1
BigIron(config-if-3/1)# exit
```

The following commands configure the Source-Active filters.

```
BigIron(config)# router msdp
BigIron(config-msdp-router)# sa-filter in 2.2.2.99
BigIron(config-msdp-router)# sa-filter in 2.2.2.97 route-map msdp_map
BigIron(config-msdp-router)# sa-filter in 2.2.2.96 route-map msdp2_map rp-route-
map msdp2_rp_map
```

The **sa-filter** commands configure the following filters:

- **sa-filter in 2.2.2.99** – This command drops all source-group pairs received from neighbor 2.2.2.99.

  NOTE: The default action is to deny all source-group pairs from the specified neighbor. If you want to permit some pairs, use route maps.

- **sa-filter in 2.2.2.97 route-map msdp_map** – This command drops source-group pairs received from neighbor 2.2.2.97 if the pairs have source address 10.x.x.x and any group address.
- **sa-filter in 2.2.2.96 route-map msdp2_map rp-route-map msdp2_rp_map** – This command accepts all source-group pairs except those associated with RP 2.2.42.3.

CLI Syntax

Syntax: [no] sa-filter in <ip-addr> [route-map <map-tag>] [rp-route-map <rp-map-tag>]

The <ip-addr> parameter specifies the IP address of the MSDP neighbor. The filter applies to Active-Source messages received from this neighbor.

The **route-map** <map-tag> parameter specifies a route map. The Layer 3 Switch applies the filter to source-group pairs that match the route map. Use the **match ip address** <acl-id> command in the route map to specify an extended ACL that contains the source and group addresses.

The **rp-route-map** <rp-map-tag> parameter specifies a route map to use for filtering based on Rendezvous Point (RP) address. Use this parameter if you want to filter Source-Active messages based on their origin. If you use the **route-map** parameter instead, messages are filtered based on source-group pairs but not based on origin. Use the **match ip route-source** <acl-id> command in the route map to specify the RP address.

NOTE: The default filter action is deny. If you want to permit some source-group pairs, use a route map. A permit action in the route map allows the Layer 3 Switch to receive the matching source-group pairs. A deny action in the route map drops the matching source-group pairs.

Filtering Advertised Source-Active Messages

The following example configures the Layer 3 Switch to advertise all source-group pairs except the ones that have source address 10.x.x.x.

Example

The following commands configure an IP address on port 3/1. This is the port on which the MSDP neighbors will be configured.

```
BigIron(config)# interface ethernet 3/1
BigIron(config-if-3/1)# ip address 2.2.2.98/24
BigIron(config-if-3/1)# exit
```

The following commands configure a loopback interface. The Layer 3 Switch will use this interface as the source address for communicating with the MSDP neighbors.

```
BigIron(config)# interface loopback 1
BigIron(config-lbif-1)# ip address 9.9.9.8/32
BigIron(config-lbif-1)# exit
```

The following command configures an extended ACL to specify the source and group addresses you want to filter.

```
BigIron(config)# access-list 123 permit 10.0.0.0 0.255.255.255 any
```

The following commands configure a route map. The map matches on source address 10.x.x.x and any group address. Since the action is deny, the Source-Active filter that uses this route map will remove the source-group pairs that match this route map from the Source-Active messages to the neighbor.

```
BigIron(config)# route-map msdp_map deny 1
BigIron(config-routemap msdp_map)# match ip address 123
BigIron(config-routemap msdp_map)# exit
```

The following commands enable MSDP and configure MSDP neighbors on port 3/1.

```
BigIron(config)# router msdp
BigIron(config-msdp-router)# exit
BigIron(config)# interface ethernet 3/1
BigIron(config-if-3/1)# msdp-peer 2.2.2.99 connect-source loopback 1
BigIron(config-if-3/1)# msdp-peer 2.2.2.97 connect-source loopback 1
BigIron(config-if-3/1)# exit
```

The following commands configure the Source-Active filter.

```
BigIron(config)# router msdp
BigIron(config-msdp-router)# sa-filter originate route-map msdp_map
```

This filter removes source-group pairs that match route map msdp_map from Source-Active messages before sending them to MSDP neighbors.

CLI Syntax

Syntax: [no] sa-filter originate [route-map <map-tag>]

The **route-map** <map-tag> parameter specifies a route map. The Layer 3 Switch applies the filter to source-group pairs that match the route map. Use the **match ip address** <acl-id> command in the route map to specify an extended ACL that contains the source and group addresses.

NOTE: The default filter action is deny. If you want to permit some source-group pairs, use a route map. A permit action in the route map allows the Layer 3 Switch to receive the matching source-group pairs. A deny action in the route map drops the matching source-group pairs.

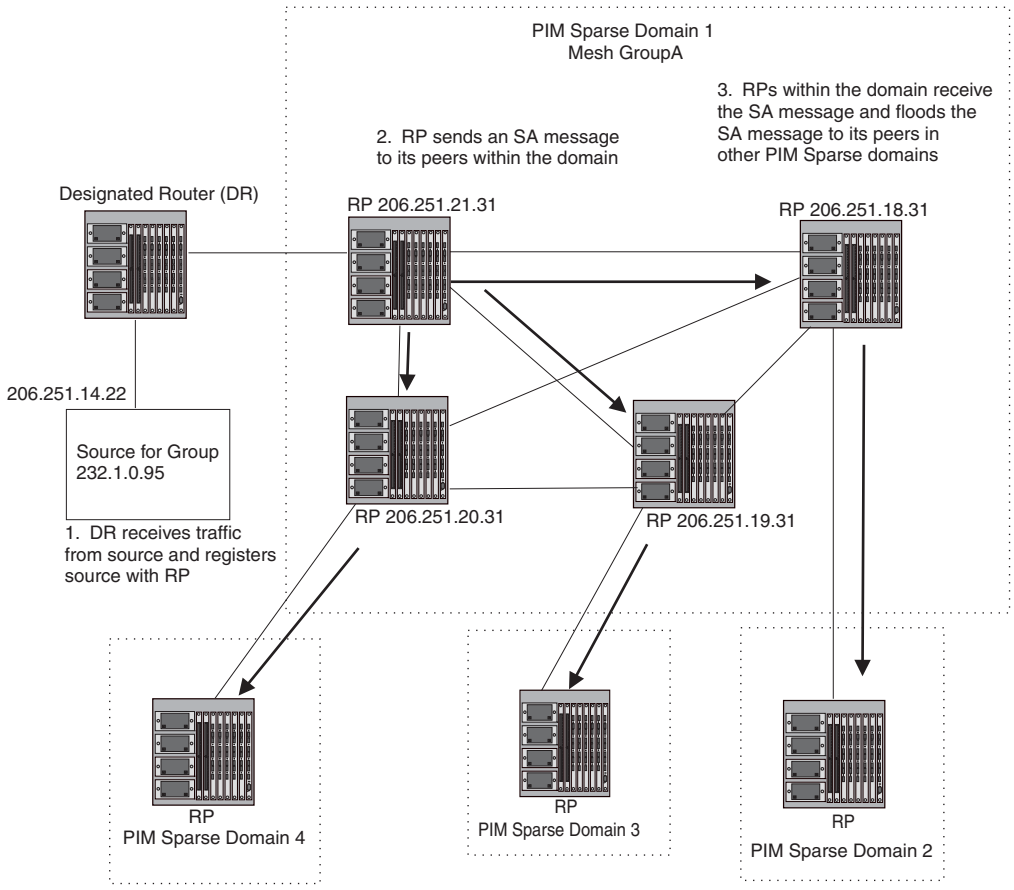
Configuring MSDP Mesh Groups

A PIM Sparse domain can have several RPs that are connected to each other to form an MSDP mesh group. To qualify as a mesh group, the RPs have to be fully meshed; that is, each RP must be connected to all peer RPs in a domain. (See Figure 14.5.)

A mesh group reduces the forwarding of SA messages within a domain. Instead of having every RP in a domain forward SA messages to all the RPs within that domain, only one RP forwards the SA message. Since an MSDP mesh group is fully meshed, peers do not forward SA messages received in a domain from one member to every member of the group. The RP that originated the SA or the first RP in a domain that receives the SA message is the only one that can forward the message to the members of a mesh group. An RP can forward an SA message to any MSRP router as long as that peer is farther away from the originating RP than the current MSRP router.

Figure 14.5 shows an example of an MSDP mesh group. In a PIM-SM mesh group the RPs are configured to be peers of each other. They can also be peers of RPs in other domains.

Figure 14.5 Example of MSDP mesh group



PIM Sparse Domain 1 in Figure 14.5 contains a mesh group with four RPs. When the first RP, for example, RP 206.251.21.41 (which is also the originating RP), receives an SA message from the source, it sends the SA message to its peers within the domain, but the peers do not send the message back to the originator RP or to each other. The RPs then send the SA message to their peers in other domains.The process continues until all RPs within the network receive the SA message. RPs send join and prune messages to appropriate points on the multicast tree towards the originating RP.

Configuring MSDP Mesh Group

To configure an MSDP mesh group, enter commands such as the following on each device that will be included in the mesh group:

```
BigIron(config)# router msdp
BigIron(config-msdp-router)# msdp-peer 163.5.34.10 connect-source loopback 2
BigIron(config-msdp-router)# msdp-peer 206.251.21.31 connect-source loopback 2
BigIron(config-msdp-router)# msdp-peer 206.251.17.31 connect-source loopback 2
BigIron(config-msdp-router)# msdp-peer 206.251.13.31 connect-source loopback 2
BigIron(config-msdp-router)# mesh-group GroupA 206.251.21.31
BigIron(config-msdp-router)# mesh-group GroupA 206.251.17.31
BigIron(config-msdp-router)# mesh-group GroupA 206.251.13.31
BigIron(config-msdp-router)# exit
```

Syntax: [no] mesh-group <group-name> <peer-address>

The sample configuration above reflects the configuration in Figure 14.5. On RP 206.251.21.31 you specify its peers within the same domain (206.251.21.31, 206.251.17.31, and 206.251.13.31).

You first configure the MSDP peers using the **msdp-peer** command to assign their IP addresses and the loopback interfaces. This information will be used as the source for sessions with the neighbor.

Next, place the MSDP peers within a domain into a mesh group. Use the **mesh-group** command. There are no default mesh groups.

The **group-name** parameter identifies the group. Enter up to 31 characters for group-name. You can have up to 4 mesh groups within a multicast network. Each mesh group can include up to 32 peers.

The **peer-address** parameter specifies the IP address of the MSDP peer that is being placed in the group.

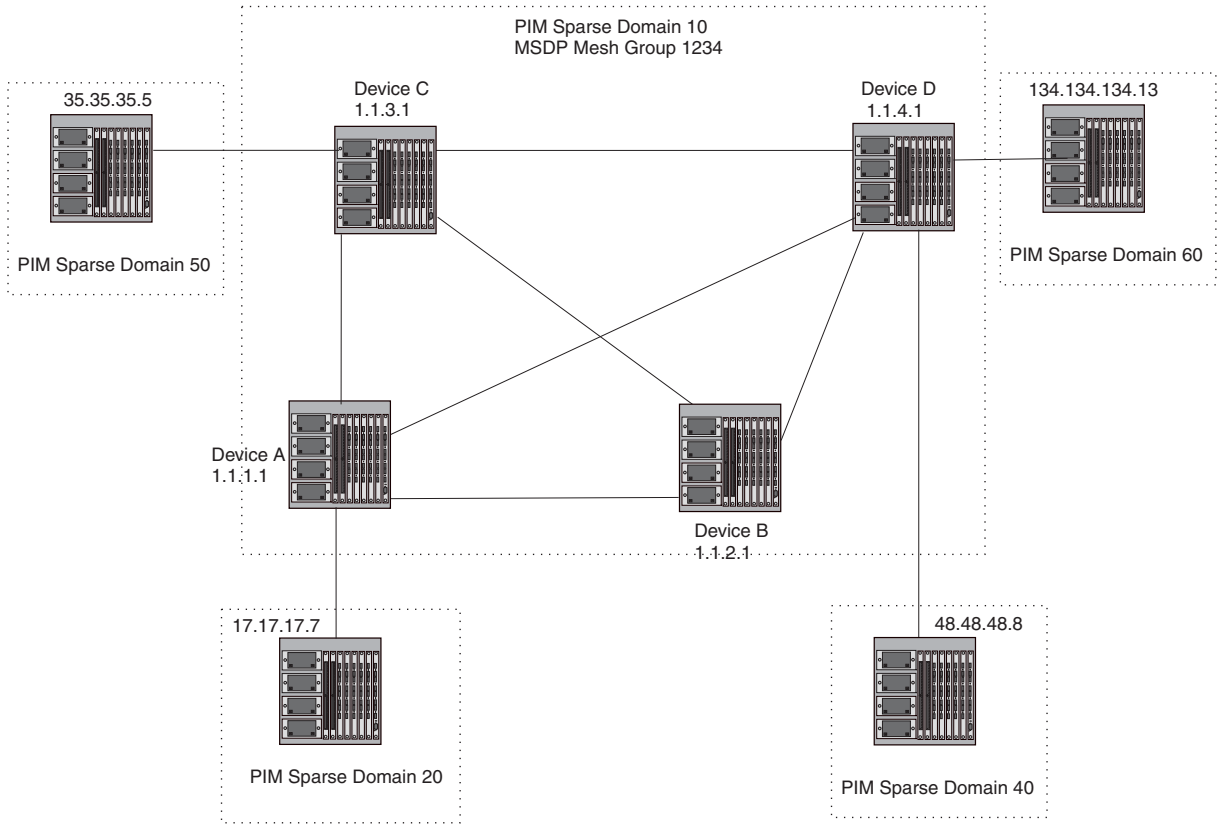
NOTE: On each of the device that will be part of the mesh-group, there must be a mesh-group definition for all the peers in the mesh-group.

Up to 32 MSDP peers can be configured per mesh group.

Example Configuration

In Figure 14.6, devices A, B, C, and D are in Mesh Group 1234. The example configuration following the figure shows how the devices are configured to be part of the MSDP mesh group. The example also shows the features that need to be enabled for the MSDP mesh group to work.

Figure 14.6 MSDP Mesh Group 1234



Configuration for Device A

The following set of commands configure the MSDP peers of Device A (1.1.1.1) that are inside and outside MSDP mesh group 1234. Device A's peers inside the mesh group 1234 are 1.1.2.1, 1.1.3.1, and 1.1.4.1. Device 17.17.17.7 is a peer of Device A, but is outside mesh group 1234. Multicast is enabled on Device A's interfaces. PIM and BGP are also enabled.

```
BigIron(config)# ip multicast-routing

BigIron(config)# ip multicast-perf

BigIron(config)# router pim

BigIron(config)# router msdp
BigIron(config-msdp-router)# msdp-peer 1.1.3.1 connect-source loopback 1
BigIron(config-msdp-router)# msdp-peer 1.1.4.1 connect-source loopback 1
BigIron(config-msdp-router)# msdp-peer 1.1.2.1 connect-source loopback 1
BigIron(config-msdp-router)# msdp-peer 17.17.17.7
BigIron(config-msdp-router)# mesh-group 1234 1.1.4.1
BigIron(config-msdp-router)# mesh-group 1234 1.1.3.1
BigIron(config-msdp-router)# mesh-group 1234 1.1.2.1
BigIron(config-msdp-router)# exit

BigIron(config)# interface loopback 1
BigIron(config-lbif-1)#ip address 1.1.1.1 255.255.255.0
BigIron(config-lbif-1)# ip pim-sparse
BigIron(config-lbif-1)# exit

BigIron(config)# interface ethernet 1/1
BigIron(config-if-1/1)# ip address 14.14.14.1 255.255.255.0
```



```
BigIron(config-if-1/1)# ip pim-sparse
BigIron(config-if-1/1)# exit

BigIron(config)# interface ethernet 2/1
BigIron(config-if-2/1)# ip address 12.12.12.1 255.255.255.0
BigIron(config-if-2/1)# ip pim-sparse
BigIron(config-if-2/1)# exit

BigIron(config)# interface ethernet 2/20
BigIron(config-if-2/20)# ip address 159.159.159.1 255.255.255.0
BigIron(config-if-2/20)# ip pim-sparse
BigIron(config-if-2/20)# exit

BigIron(config)# interface ethernet 4/1
BigIron(config-if-4/1)# ip address 31.31.31.1 255.255.255.0
BigIron(config-if-4/1)# ip pim-sparse
BigIron(config-if-4/1)# exit

BigIron(config)# interface ethernet 4/8
BigIron(config-if-4/8)# ip address 17.17.17.1 255.255.255.0
BigIron(config-if-4/8)# ip pim-sparse
BigIron(config-if-4/8)# ip pim border
BigIron(config-if-4/8)# exit

BigIron(config)# router pim
BigIron(config-router-pim)# bsr-candidate loopback 1 1 31
BigIron(config-router-pim)# rp-candidate loopback 1
BigIron(config-router-pim)# exit

BigIron(config)# router bgp
BigIron(config-bgp-router)# local-as 111
BigIron(config-bgp-router)# neighbor 31.31.31.3 remote-as 333
BigIron(config-bgp-router)# neighbor 31.31.31.3 next-hop-self
BigIron(config-bgp-router)# neighbor 12.12.12.2 remote-as 222
BigIron(config-bgp-router)# neighbor 12.12.12.2 next-hop-self
BigIron(config-bgp-router)# neighbor 14.14.14.4 remote-as 444
BigIron(config-bgp-router)# neighbor 14.14.14.4 next-hop-self
BigIron(config-bgp-router)# neighbor 17.17.17.7 remote-as 777
BigIron(config-bgp-router)# neighbor 17.17.17.7 next-hop-self
BigIron(config-bgp-router)# redistribute connected
BigIron(config-bgp-router)# write memory
```

Configuration for Device B

The following set of commands configure the MSDP peers of Device B. All Device B's peers (1.1.1.1, 1.1.3.1, and 1.1.4.1) are in the MSDP mesh group 1234. Multicast is enabled on Device B's interfaces. PIM and BGP are also enabled.

```
BigIron(config)# ip multicast-routing

BigIron(config)# ip multicast-perf

BigIron(config)# router pim

BigIron(config)# router msdp
BigIron(config-msdp-router)# msdp-peer 1.1.3.1 connect-source loopback 1
BigIron(config-msdp-router)# msdp-peer 1.1.1.1 connect-source loopback 1
BigIron(config-msdp-router)# msdp-peer 1.1.4.1 connect-source loopback 1
BigIron(config-msdp-router)# mesh-group 1234 1.1.1.1
BigIron(config-msdp-router)# mesh-group 1234 1.1.3.1
BigIron(config-msdp-router)# mesh-group 1234 1.1.4.1
BigIron(config-msdp-router)# exit

BigIron(config)# interface loopback 1
BigIron(config-lbif-1)# ip address 1.1.2.1 255.255.255.0
```



```
BigIron(config-lbif-1)# ip pim-sparse
BigIron(config-lbif-1)# exit

BigIron(config)# interface ethernet 1/1
BigIron(config-if-1/1)# ip address 12.12.12.2 255.255.255.0
BigIron(config-if-1/1)# ip pim-sparse
BigIron(config-if-1/1)# exit

BigIron(config)# interface ethernet 1/12
BigIron(config-if-1/12)# ip address 165.165.165.1 255.255.255.0
BigIron(config-if-1/12)# ip pim-sparse
BigIron(config-if-1/12)# exit

BigIron(config)# interface ethernet 1/24
BigIron(config-if-1/24)# ip address 168.72.2.2 255.255.255.0
BigIron(config-if-1/24)# exit

BigIron(config)# interface ethernet 1/25
BigIron(config-if-1/25)# ip address 24.24.24.2 255.255.255.0
BigIron(config-if-1/25)# ip pim-sparse
BigIron(config-if-1/24)# exit

BigIron(config)# interface ethernet 8/1
BigIron(config-if-8/1)# ip address 32.32.32.2 255.255.255.0
BigIron(config-if-8/1)# ip pim-sparse
BigIron(config-if-1/24)# exit

BigIron(config)# router pim
BigIron(config-router-pim)# bsr-candidate loopback 1 2 32
BigIron(config-router-pim)# rp-candidate loopback 1
BigIron(config-router-pim)# exit

BigIron(config)# router bgp
BigIron(config-router-bgp)# local-as 222
BigIron(config-router-bgp)# neighbor 32.32.32.3 remote-as 333
BigIron(config-router-bgp)# neighbor 32.32.32.3 next-hop-self
BigIron(config-router-bgp)# neighbor 24.24.24.4 remote-as 444
BigIron(config-router-bgp)# neighbor 24.24.24.4 next-hop-self
BigIron(config-router-bgp)# neighbor 12.12.12.1 remote-as 111
BigIron(config-router-bgp)# neighbor 12.12.12.1 next-hop-self
BigIron(config-router-bgp)# redistribute connected
BigIron(config-router-bgp)# write memory
```

Configuration for Device C

The following set of commands configure the MSDP peers of Device C (1.1.3.1) that are inside and outside MSDP mesh group 1234. Device C's peers inside the mesh group 1234 are 1.1.1.1, 1.1.2.1, and 1.1.4.1. Device 35.35.35.5 is a peer of Device C, but is outside mesh group 1234. . Multicast is enabled on Device C's interfaces. PIM and BGP are also enabled.

```
BigIron(config)# ip multicast-routing

BigIron(config)# ip multicast-perf

BigIron(config)# router pim

BigIron(config)# router msdp
BigIron(config-msdp-router)# msdp-peer 35.35.35.5
BigIron(config-msdp-router)# msdp-peer 1.1.2.1 connect-source loopback 1
BigIron(config-msdp-router)# msdp-peer 1.1.4.1 connect-source loopback 1
BigIron(config-msdp-router)# msdp-peer 1.1.1.1 connect-source loopback 1
BigIron(config-msdp-router)# mesh-group 1234 1.1.2.1
BigIron(config-msdp-router)# mesh-group 1234 1.1.1.1
BigIron(config-msdp-router)# mesh-group 1234 1.1.4.1
BigIron(config-msdp-router)# exit
```



```
BigIron(config)# interface loopback 1
BigIron(config-lbif-1)# ip address 1.1.3.1 255.255.255.0
BigIron(config-lbif-1)# ip pim-sparse
BigIron(config-lbif-1)# exit

BigIron(config)# interface ethernet 3/1
BigIron(config-if-3/1)# ip address 32.32.32.3 255.255.255.0
BigIron(config-if-3/1)# ip pim-sparse
BigIron(config-if-3/1)# exit

BigIron(config)# interface ethernet 10/1
BigIron(config-if-10/1)# ip address 31.31.31.3 255.255.255.0
BigIron(config-if-10/1)# ip pim-sparse
BigIron(config-if-10/1)# exit

BigIron(config)# interface ethernet 10/8
BigIron(config-if-10/8)# ip address 35.35.35.3 255.255.255.0
BigIron(config-if-10/8)# ip pim-sparse
BigIron(config-if-10/8)# ip pim border
BigIron(config-if-10/8)# exit

BigIron(config)# interface ethernet 12/2
BigIron(config-if-12/1)# ip address 34.34.34.3 255.255.255.0
BigIron(config-if-12/1)# ip pim-sparse
BigIron(config-if-12/1)# exit

BigIron(config)# interface ethernet 14/4
BigIron(config-if-14/4)# ip address 154.154.154.1 255.255.255.0
BigIron(config-if-12/1)# ip pim-sparse
BigIron(config-if-12/1)# exit

BigIron(config)# router pim
BigIron(config-router-pim)# bsr-candidate loopback 1 1 3
BigIron(config-router-pim)# rp-candidate loopback 1
BigIron(config-router-pim)# exit

BigIron(config)# router bgp
BigIron(config-router-bsr)# local-as 333
BigIron(config-router-bsr)# neighbor 35.35.35.5 remote-as 555
BigIron(config-router-bsr)# neighbor 35.35.35.5 next-hop-self
BigIron(config-router-bsr)# neighbor 32.32.32.2 remote-as 222
BigIron(config-router-bsr)# neighbor 32.32.32.2 next-hop-self
BigIron(config-router-bsr)# neighbor 34.34.34.4 remote-as 444
BigIron(config-router-bsr)# neighbor 34.34.34.4 next-hop-self
BigIron(config-router-bsr)# neighbor 31.31.31.1 remote-as 111
BigIron(config-router-bsr)# neighbor 31.31.31.1 next-hop-self
BigIron(config-router-bsr)# redistribute connected
BigIron(config-router-bsr)# write memory
```

Configuration for Device D

The following set of commands configure the MSDP peers of Device D (1.1.4.1) that are inside and outside MSDP mesh group 1234. Device D's peers inside the mesh group 1234 are 1.1.1.1, 1.1.2.1, and 1.1.3.1. Device 48.48.48.8 and 134.134.134.13 are also peers of Device D, but are outside mesh group 1234. . Multicast is enabled on Device D's interfaces. PIM and BGP are also enabled.

```
BigIron(config)# ip multicast-routing

BigIron(config)# ip multicast-perf

BigIron(config)# router pim

BigIron(config)# router msdp
BigIron(config-msdp-router)# msdp-peer 1.1.3.1 connect-source loopback 1
BigIron(config-msdp-router)# msdp-peer 1.1.1.1 connect-source loopback 1
```



```
BigIron(config-msdp-router)# msdp-peer 1.1.2.1 connect-source loopback 1
BigIron(config-msdp-router)# msdp-peer 48.48.48.8
BigIron(config-msdp-router)# msdp-peer 134.134.134.13
BigIron(config-msdp-router)# mesh-group 1234 1.1.1.1
BigIron(config-msdp-router)# mesh-group 1234 1.1.3.1
BigIron(config-msdp-router)# mesh-group 1234 1.1.2.1
BigIron(config-msdp-router)# exit

BigIron(config)# interface loopback 1
BigIron(config-lbif-)# ip address 1.1.4.1 255.255.255.0
BigIron(config-lbif-)# ip pim-sparse
BigIron(config-lbif-)# exit

BigIron(config)# interface ethernet 1/1
BigIron(config-if-)# ip address 24.24.24.4 255.255.255.0
BigIron(config-if-)# ip pim-sparse
BigIron(config-if-)# exit

BigIron(config)# interface ethernet 2/6
BigIron(config-if-)# ip address 156.156.156.1 255.255.255.0
BigIron(config-if-)# ip pim-sparse
BigIron(config-if-)# exit

BigIron(config)# interface ethernet 5/1
BigIron(config-if-)# ip address 34.34.34.4 255.255.255.0
BigIron(config-if-)# ip pim-sparse
BigIron(config-if-)# exit

BigIron(config)# interface ethernet 7/1
BigIron(config-if-)# ip address 14.14.14.4 255.255.255.0
BigIron(config-if-)# ip pim-sparse
BigIron(config-if-)# exit

BigIron(config)# interface ethernet 7/7
BigIron(config-if-)# ip address 48.48.48.4 255.255.255.0
BigIron(config-if-)# ip pim-sparse
BigIron(config-if-)# ip pim border
BigIron(config-if-)# exit

BigIron(config)# interface ethernet 7/8
BigIron(config-if-)# ip address 134.134.134.4 255.255.255.0
BigIron(config-if-)# ip pim-sparse
BigIron(config-if-)# ip pim border
BigIron(config-if-)# exit

BigIron(config)# router pim
BigIron(config-router-pim)# bsr-candidate loopback 1 14 34
BigIron(config-router-pim)# rp-candidate loopback 1
BigIron(config-router-pim)# exit

BigIron(config)# router bgp
BigIron(config-router-bsr)# local-as 444
BigIron(config-router-bsr)# neighbor 34.34.34.3 remote-as 333
BigIron(config-router-bsr)# neighbor 34.34.34.3 next-hop-self
BigIron(config-router-bsr)# neighbor 14.14.14.1 remote-as 111
BigIron(config-router-bsr)# neighbor 14.14.14.1 next-hop-self
BigIron(config-router-bsr)# neighbor 24.24.24.2 remote-as 222
BigIron(config-router-bsr)# neighbor 24.24.24.2 next-hop-self
BigIron(config-router-bsr)# neighbor 48.48.48.8 remote-as 888
BigIron(config-router-bsr)# neighbor 48.48.48.8 next-hop-self
BigIron(config-router-bsr)# neighbor 134.134.134.13 remote-as 1313
BigIron(config-router-bsr)# neighbor 134.134.134.13 next-hop-self
BigIron(config-router-bsr)# redistribute connected
```



```
BigIron(config-router-bsr)# write memory
```

Displaying MSDP Information

You can display the following MSDP information:

- Summary information – the IP addresses of the peers, the state of the Layer 3 Switch's MSDP session with each peer, and statistics for Keepalive, Source Active, and Notification messages sent to and received from each of the peers
- Peer information – the IP address of the peer, along with detailed MSDP and TCP statistics
- Source Active cache entries – the Source Active messages cached by the Layer 3 Switch

Displaying Summary Information

To display summary MSDP information, use the following CLI method.

USING THE CLI

To display summary MSDP information, enter the following command at any level of the CLI:

```
BigIron(config-msdp-router)# show ip msdp summary

MSDP Peer Status Summary
 KA: Keepalive SA:Source-Active NOT: Notification
 Peer Address     State         KA          SA          NOT
                              In     Out   In    Out   In   Out
 206.251.17.30    ESTABLISH   3      3     0     640   0     0
 206.251.17.41    ESTABLISH   0      3     651   0     0     0
```

Syntax: show ip msdp summary

This display shows the following information.

Table 14.2: MSDP Summary Information

This Field...	Displays...
Peer Address	The IP address of the peer's interface with the Layer 3 Switch
State	The state of the MSDP router's connection with the peer. The state can be one of the following: • CONNECTING – The session is in the active open state. • ESTABLISHED – The MSDP session is fully up. • INACTIVE – The session is idle. • LISTENING – The session is in the passive open state.
KA In	The number of MSDP Keepalive messages the MSDP router has received from the peer
KA Out	The number of MSDP Keepalive messages the MSDP router has sent to the peer
SA In	The number of Source Active messages the MSDP router has received from the peer
SA Out	The number of Source Active messages the MSDP router has sent to the peer

Table 14.2: MSDP Summary Information (Continued)

This Field...	Displays...
NOT In	The number of Notification messages the MSDP router has received from the peer
NOT Out	The number of Notification messages the MSDP router has sent to the peer

USING THE WEB MANAGEMENT INTERFACE

You cannot display MSDP information using the Web management interface.

Displaying Peer Information

To display summary MSDP peer information, use the following CLI method.

USING THE CLI

To display MSDP peer information, use the following CLI method.

```
BigIron(config-msdp-router)# show ip msdp peer

        Total number of MSDP Peers: 2

        IP Address           State
1       206.251.17.30        ESTABLISHED
        Keep Alive Time  Hold Time
        60               90

                             Message Sent         Message Received
        Keep Alive           2                    3
        Notifications        0                    0
        Source-Active        0                    640
        Last Connection Reset Reason:Reason Unknown
        Notification Message Error Code Received:Unspecified
        Notification Message Error SubCode Received:Not Applicable
        Notification Message Error Code Transmitted:Unspecified
        Notification Message Error SubCode Transmitted:Not Applicable
        TCP Connection state: ESTABLISHED
          Local host:  206.251.17.29, Local  Port: 8270
          Remote host: 206.251.17.30, Remote Port: 639
          ISentSeq:      16927  SendNext:     685654  TotUnAck:          0
          SendWnd:       16384  TotSent:      668727  ReTrans:           1
          IRcvSeq:    45252428  RcvNext:    45252438  RcvWnd:        16384
          TotalRcv:         10  RcvQue:            0  SendQue:           0
```

Syntax: show ip msdp peer

This display shows the following information.

Table 14.3: MSDP Peer Information

This Field...	Displays...
Total number of MSDP peers	The number of MSDP peers configured on the Layer 3 Switch

Table 14.3: MSDP Peer Information (Continued)

This Field...	Displays...
IP Address	The IP address of the peer's interface with the Layer 3 Switch
State	The state of the MSDP router's connection with the peer. The state can be one of the following: • CONNECTING – The session is in the active open state. • ESTABLISHED – The MSDP session is fully up. • INACTIVE – The session is idle. • LISTENING – The session is in the passive open state.
Keep Alive Time	The keep alive time, which specifies how often this MSDP router sends keep alive messages to the neighbor. The keep alive time is 60 seconds and is not configurable.
Hold Time	The hold time, which specifies how many seconds the MSDP router will wait for a KEEPALIVE or UPDATE message from an MSDP neighbor before deciding that the neighbor is dead. The hold time is 90 seconds and is not configurable.
Keep Alive Message Sent	The number of Keep Alive messages the MSDP router has sent to the peer.
Keep Alive Message Received	The number of Keep Alive messages the MSDP router has received from the peer.
Notifications Sent	The number of Notification messages the MSDP router has sent to the peer.
Notifications Received	The number of Notification messages the MSDP router has received from the peer.
Source-Active Sent	The number of Source Active messages the MSDP router has sent to the peer.
Source-Active Received	The number of Source Active messages the MSDP router has received from the peer.
Last Connection Reset Reason	The reason the previous session with this neighbor ended.

Table 14.3: MSDP Peer Information (Continued)

This Field...	Displays...
Notification Message Error Code Received	If the MSDP router receives a NOTIFICATION messages 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. • 1 – Message Header Error • 2 – SA-Request Error • 3 – SA-Message/SA-Response Error • 4 – Hold Timer Expired • 5 – Finite State Machine Error • 6 – Notification • 7 – Cease For information about these error codes, see section 17 in the Internet draft describing MSDP, “draft-ietf-msdp-spec”.
Notification Message Error SubCode Received	See above.
Notification Message Error Code Transmitted	The error message corresponding to the error code in the NOTIFICATION message this MSDP router sent to the neighbor. See the description for the Notification Message Error Code Received field for a list of possible codes.
Notification Message Error SubCode Transmitted	See above.
TCP Statistics	

Table 14.3: MSDP Peer 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.
Local host	The IP address of the MSDP router's interface with the peer.
Local port	The TCP port the MSDP router is using for the BGP4 TCP session with the neighbor.
Remote host	The IP address of the neighbor.
Remote port	The TCP port number of the peer end of the connection.
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 MSDP router that have not been acknowledged by the neighbor.
SendWnd	The size of the send window.
TotSent	The number of sequence numbers sent to the neighbor.
ReTrans	The number of sequence numbers that the MSDP router retransmitted because they were not acknowledged.

Table 14.3: MSDP Peer Information (Continued)

This Field...	Displays...
IRcvSeq	The initial receive sequence number for the session.
RcvNext	The next sequence number expected from the neighbor.
RcvWnd	The size of the receive window.
TotalRcv	The number of sequence numbers received from the neighbor.
RcvQue	The number of sequence numbers in the receive queue.
SendQue	The number of sequence numbers in the send queue.

USING THE WEB MANAGEMENT INTERFACE

You cannot display MSDP information using the Web management interface.

Displaying Source Active Cache Information

To display the Source Actives in the MSDP cache, use the following CLI method.

```
BigIron(config-msdp-router)# show ip msdp sa-cache

Total Entry 4096, Used 1800 Free 2296
Index   SourceAddr  GroupAddr              Age
1     (100.100.1.254, 232.1.0.95), RP:206.251.17.41, Age:0
2     (100.100.1.254, 237.1.0.98), RP:206.251.17.41, Age:30
3     (100.100.1.254, 234.1.0.48), RP:206.251.17.41, Age:30
4     (100.100.1.254, 239.1.0.51), RP:206.251.17.41, Age:30
5     (100.100.1.254, 234.1.0.154), RP:206.251.17.41, Age:30
6     (100.100.1.254, 236.1.0.1), RP:206.251.17.41, Age:30
7     (100.100.1.254, 231.1.0.104), RP:206.251.17.41, Age:90
8     (100.100.1.254, 239.1.0.157), RP:206.251.17.41, Age:30
9     (100.100.1.254, 236.1.0.107), RP:206.251.17.41, Age:30
10    (100.100.1.254, 233.1.0.57), RP:206.251.17.41, Age:90
```

Syntax: show ip msdp sa-cache

This display shows the following information.

Table 14.4: MSDP Source Active Cache

This Field...	Displays...
Total Entry	The total number of entries the cache can hold.
Used	The number of entries the cache currently contains.
Free	The number of additional entries for which the cache has room.
Index	The cache entry number.
SourceAddr	The IP address of the multicast source.
GroupAddr	The IP multicast group to which the source is sending information.

Table 14.4: MSDP Source Active Cache (Continued)

This Field...	Displays...
RP	The RP through which receivers can access the group traffic from the source
Age	The number of seconds the entry has been in the cache

USING THE WEB MANAGEMENT INTERFACE

You cannot display MSDP information using the Web management interface.

Clearing MSDP Information

You can clear the following MSDP information:

- Peer information
- Source Active cache
- MSDP statistics

Clearing Peer Information

To clear MSDP peer information, enter the following command at the Privileged EXEC level of the CLI:

```
BigIron# clear ip msdp peer 205.216.162.1
Remote connection closed
```

Syntax: clear ip msdp peer <ip-addr>

The command in this example clears the MSDP peer connection with MSDP router 205.216.162.1. The CLI displays a message to indicate when the connection has been successfully closed.

Clearing the Source Active Cache

To clear the entries from the Source Active cache, enter the following command at the Privileged EXEC level of the CLI:

```
BigIron# clear ip msdp sa-cache
```

Syntax: clear ip msdp sa-cache [<source-addr> | <group-addr>]

The command in this example clears all the cache entries. Use the <source-addr> parameter to clear only the entries for a specified course. Use the <group-addr> parameter to clear only the entries for a specific group.

Clearing MSDP Statistics

To clear MSDP statistics, enter the following command at the Privileged EXEC level of the CLI:

```
BigIron# clear ip msdp statistics
```

Syntax: clear ip msdp statistics [<ip-addr>]

The command in this example clears statistics for all the peers. To clear statistics for only a specific peer, enter the peer's IP address.

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 14.7. 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 14.7, 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 14.8, 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 14.7 Downstream broadcast of IP multicast packets from source host

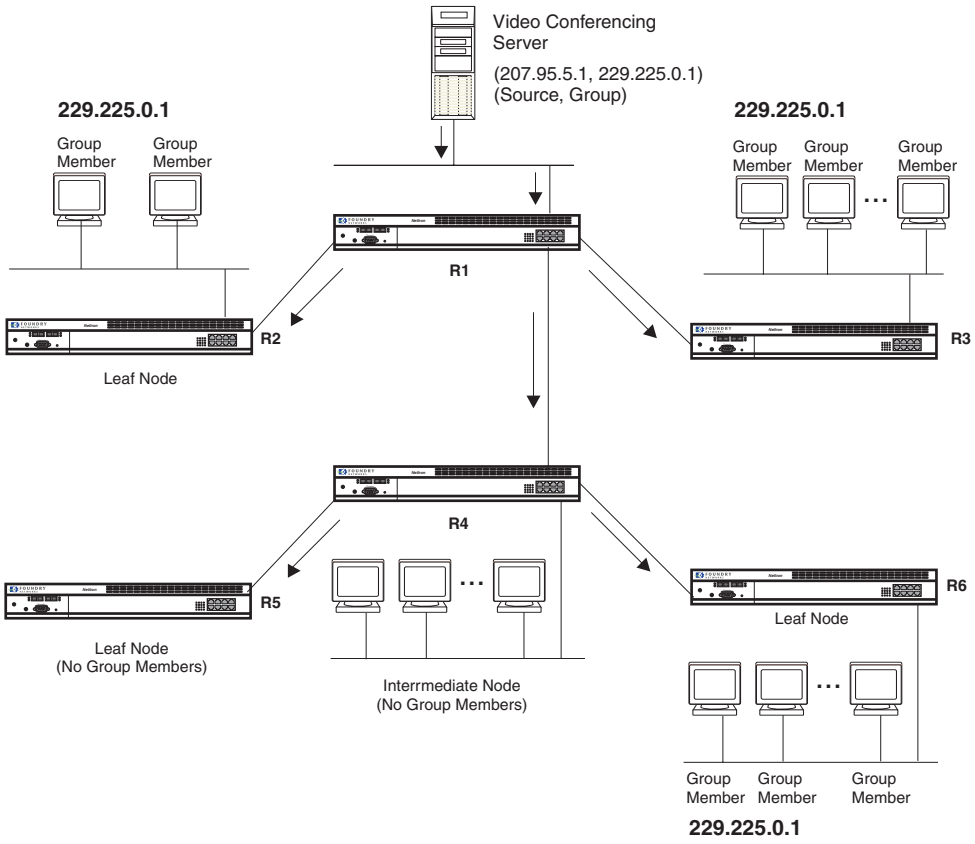
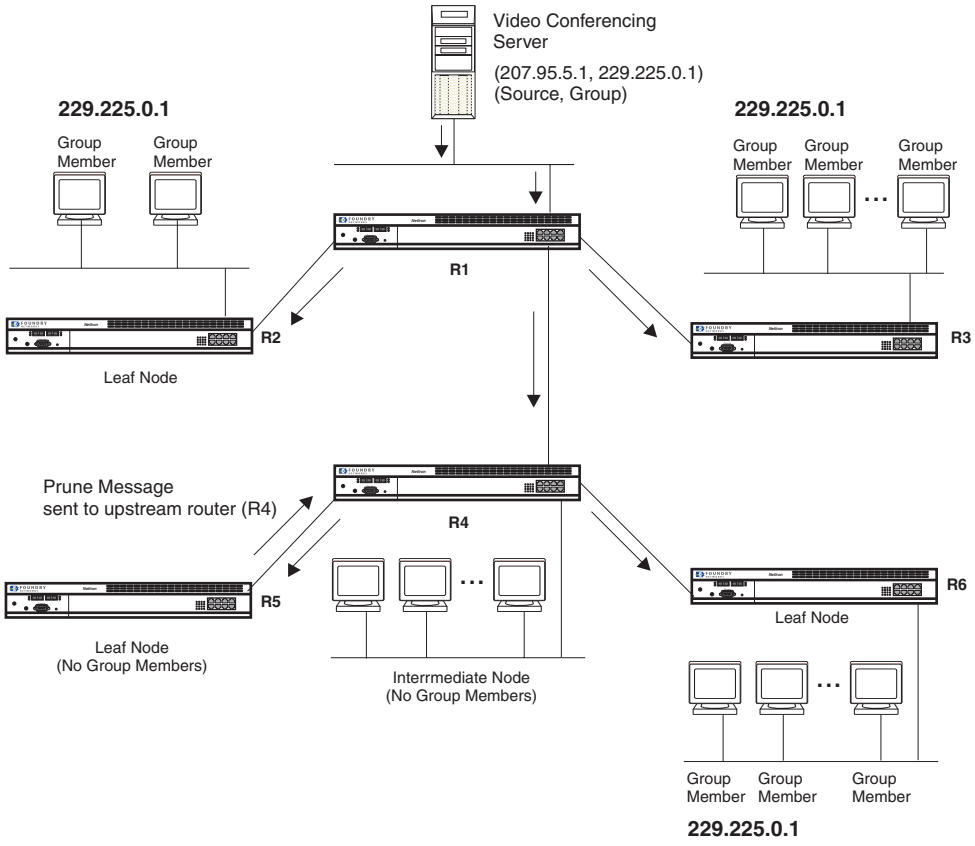


Figure 14.8 Pruning leaf nodes from a multicast tree



Grafts to a Multicast Tree

A DVMRP router restores pruned branches to a multicast tree by sending graft messages towards the upstream router. Graft messages start at the leaf node and travel up the tree, first sending the message to its neighbor upstream router.

In the example above, if a new 229.255.0.1 group member joins on router R6, which had been pruned previously, a graft will be sent upstream to R4. Since the forwarding state for this entry is in a prune state, R4 sends a graft to R1. Once R4 has joined the tree, it along with R6 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 14.7.

DVMRP is enabled on each of the Foundry Layer 3 Switches shown in Figure 14.7, 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 router.

Globally Enabling and Disabling DVMRP

To globally enable DVMRP, enter the following command:

```
Router1(config)# router dvmrp
```

Syntax: [no] router dvmrp

Prior to software release 07.8.00, the behavior of the **[no] router dvmrp** command was as follows:

- Foundry Layer 3 Switches required a software reload whenever you enabled DVMRP using the **router dvmrp** command.
- Entering a **no router dvmrp** command removed all configuration for PIM multicast on a Layer 3 Switch (**router pim** level) and all PIM and PIM-Sparse (**ip pim** and **ip pim-sparse)** configuration on all interfaces.

Beginning with software release 07.8.00:

- 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:

```
BigIron(config)# router dvmrp
BigIron(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.

USING THE CLI

To enable DVMRP on Router 1 and interface 3, enter the following:

```
Router1(config)# router dvmrp
Router1(config-dvmrp-router)# int e 3
Router1(config-if-3)# ip dvmrp
```

USING THE WEB MANAGEMENT INTERFACE

To enable DVMRP on Router 1 and interface 3, enter the following:

1. Log on to the device using a valid user name and password for read-write access.
2. If you have not already enabled DVMRP, enable it by clicking on the Enable radio button next to DVMRP on the System configuration panel, then clicking Apply to apply the change.
3. Click on the plus sign next to Configure in the tree view to expand the list of configuration options.
4. Click on the plus sign next to DVMRP in the tree view to expand the list of DVMRP option links.
5. Click on the Virtual Interface link to display the DVMRP Interface configuration panel.

NOTE: If the device already has DVMRP interfaces, a table listing the interfaces is displayed. Click the Modify button to the right of the row describing an interface to change its configuration, or click the Add Virtual Interface link to display the DVMRP Interface configuration panel.

---

6. Select the interface type. You can select Subnet or Tunnel.
7. Select the IP address of the interface being configured from the Local Address pulldown menu.

8. If you are configuring an IP Tunnel, enter the IP address of the destination interface, the end point of the IP Tunnel, in the Remote Address field. IP tunneling must also be enabled and defined on the destination router interface as well.

   NOTE: The Remote Address field applies only to tunnel interfaces, not to subnet interfaces.

9. Modify the time to live threshold (TTL) if necessary. The TTL defines the minimum value required in a packet in order for the packet to be forwarded out the interface.

   NOTE: 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 values are 1 – 64. The default value is 1.

10. Click Enable or Disable next to Advertise Local to enable or disable the feature.
11. Click Enable or Disable next to Encapsulation to enable or disable the feature.
12. Click the Add button to save the change to the device's running-config file.
13. 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.
14. Click on the plus sign next to Command in the tree view to list the command options.
15. Select the Reload link and select Yes when prompted to reload the software. You must reload after enabling DVMRP to place the change into effect. If DVMRP was already enabled when you added the interface, you do not need to reload.

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.

USING THE CLI

To modify the neighbor timeout value to 100, enter the following:

```
BigIron(config-dvmrp-router)# nbr 100
```

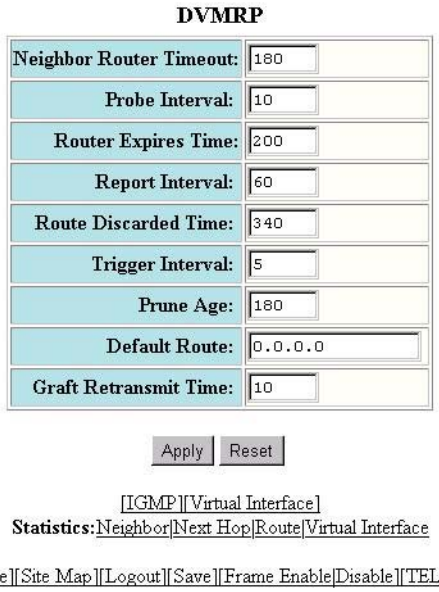
Syntax: nbr-timeout <40-8000>

The default is 180 seconds.

USING THE WEB MANAGEMENT INTERFACE

1. Log on to the device using a valid user name and password for read-write access.

2. Click on the plus sign next to Configure in the tree view to expand the list of configuration options.
3. Click on the plus sign next to DVMRP in the tree view to expand the list of DVMRP option links.
4. Click on the General link to display the DVMRP configuration panel, as shown in the following example.



5. Enter a value from 40 – 8000 into the Neighbor Router Timeout field.
6. Click the Apply button to save the change to the device's running-config file.
7. 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.

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.

USING THE CLI

To modify the route expire setting to 50, enter the following:

```
BigIron(config-dvmrp-router)# route-expire-timeout 50
```

Syntax: route-expire-timeout <20-4000>

USING THE WEB MANAGEMENT INTERFACE

1. Log on to the device using a valid user name and password for read-write access.
2. Click on the plus sign next to Configure in the tree view to expand the list of configuration options.
3. Click on the plus sign next to DVMRP in the tree view to expand the list of DVMRP option links.
4. Click on the General link to display the DVMRP configuration panel.
5. Enter a value from 20 – 4000 in the Route Expire Time field.
6. Click the Apply button to save the change to the device's running-config file.
7. 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.

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.

USING THE CLI

To modify the route discard setting to 150, enter the following:

```
BigIron(config-dvmrp-router)# route-discard-timeout 150
```

Syntax: route-discard-timeout <40-8000>

USING THE WEB MANAGEMENT INTERFACE

1. Log on to the device using a valid user name and password for read-write access.
2. Click on the plus sign next to Configure in the tree view to expand the list of configuration options.
3. Click on the plus sign next to DVMRP in the tree view to expand the list of DVMRP option links.
4. Click on the General link to display the DVMRP configuration panel.
5. Enter a value from 40 – 8000 in the Route Discard Time field.
6. Click the Apply button to save the change to the device's running-config file.
7. 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.

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.

USING THE CLI

To modify the prune age setting to 150, enter the following:

```
BigIron(config-dvmrp-router)# prune 25
```

Syntax: prune-age <20-3600>

USING THE WEB MANAGEMENT INTERFACE

1. Log on to the device using a valid user name and password for read-write access.
2. Click on the plus sign next to Configure in the tree view to expand the list of configuration options.
3. Click on the plus sign next to DVMRP in the tree view to expand the list of DVMRP option links.
4. Click on the General link to display the DVMRP configuration panel.
5. Enter a value from 20 – 3600 in the Prune Age field.
6. Click the Apply button to save the change to the device's running-config file.
7. 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.

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.

USING THE CLI

To modify the setting for graft retransmit time to 120, enter the following:

```
BigIron(config-dvmrp-router)# graft 120
```


Syntax: graft-retransmit-time <5-3600>

USING THE WEB MANAGEMENT INTERFACE

1. Log on to the device using a valid user name and password for read-write access.
2. Click on the plus sign next to Configure in the tree view to expand the list of configuration options.
3. Click on the plus sign next to DVMRP in the tree view to expand the list of DVMRP option links.
4. Click on the General link to display the DVMRP configuration panel.
5. Enter a value from 5 – 3600 in the Graft Retransmit Time field.
6. Click the Apply button to save the change to the device's running-config file.
7. 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.

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.

USING THE CLI

To modify the probe interval setting to 10, enter the following:

```
BigIron(config-dvmrp-router)# probe 10
```

Syntax: probe-interval <5-30>

USING THE WEB MANAGEMENT INTERFACE

1. Log on to the device using a valid user name and password for read-write access.
2. Click on the plus sign next to Configure in the tree view to expand the list of configuration options.
3. Click on the plus sign next to DVMRP in the tree view to expand the list of DVMRP option links.
4. Click on the General link to display the DVMRP configuration panel.
5. Enter a value from 5 – 30 in the Probe Interval field.
6. Click the Apply button to save the change to the device's running-config file.
7. 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.

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.

USING THE CLI

To support propagation of DVMRP routing information to the network every 90 seconds, enter the following:

```
BigIron(config-dvmrp-router)# report 90
```

Syntax: report-interval <10-2000>

USING THE WEB MANAGEMENT INTERFACE

1. Log on to the device using a valid user name and password for read-write access.
2. Click on the plus sign next to Configure in the tree view to expand the list of configuration options.
3. Click on the plus sign next to DVMRP in the tree view to expand the list of DVMRP option links.
4. Click on the General link to display the DVMRP configuration panel.
5. Enter a value from 10 – 2000 in the Report Interval field.

6. Click the Apply button to save the change to the device's running-config file.
7. 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.

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.

USING THE CLI

To support the sending of trigger updates every 20 seconds, enter the following:

```
BigIron(config-dvmrp-router)# trigger-interval 20
```

Syntax: trigger-interval <5-30>

USING THE WEB MANAGEMENT INTERFACE

1. Log on to the device using a valid user name and password for read-write access.
2. Click on the plus sign next to Configure in the tree view to expand the list of configuration options.
3. Click on the plus sign next to DVMRP in the tree view to expand the list of DVMRP option links.
4. Click on the General link to display the DVMRP configuration panel.
5. Enter a value from 5 – 30 in the Trigger Interval field.
6. Click the Apply button to save the change to the device's running-config file.
7. 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.

Modifying Default Route

This defines the default gateway for IP multicast routing.

USING THE CLI

To define the default gateway for DVMRP, enter the following:

```
BigIron(config-dvmrp-router)# default-gateway 192.35.4.1
```

Syntax: default-gateway <ip-addr>

USING THE WEB MANAGEMENT INTERFACE

1. Log on to the device using a valid user name and password for read-write access.
2. Click on the plus sign next to Configure in the tree view to expand the list of configuration options.
3. Click on the plus sign next to DVMRP in the tree view to expand the list of DVMRP option links.
4. Click on the General link to display the DVMRP configuration panel.
5. Enter the IP address of the default gateway in the Default Route field.
6. Click the Apply button to save the change to the device's running-config file.
7. 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.

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.

USING THE CLI

To set a TTL of 64, enter the following:

```
BigIron(config)# int e 1/4
BigIron(config-if-1/4)# ip dvmrp ttl 60
```

Syntax: ttl-threshold <1-64>

USING THE WEB MANAGEMENT INTERFACE

To modify a DVMRP interface's TTL:

1. Log on to the device using a valid user name and password for read-write access.
2. Click on the plus sign next to Configure in the tree view to expand the list of configuration options.
3. Click on the plus sign next to DVMRP in the tree view to expand the list of DVMRP option links.
4. Select the Virtual Interface link to display a table listing the configured DVMRP Interfaces.
5. Click on the Modify button next to the interface you want to modify. The DVMRP Interface configuration panel is displayed.
6. Enter a value from 1 – 64 in the Time To Live Threshold (TTL) field.
7. Click the Add button to save the changes to the device's running-config file.
8. 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.

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.

USING THE CLI

To set a metric of 15 for a DVMRP interface, enter the following:

```
BigIron(config)# interface 3/5
BigIron(config-if-3/5)# ip dvmrp metric 15
```

Syntax: ip dvmrp metric <1-31>

USING THE WEB MANAGEMENT INTERFACE

To modify a DVMRP interface's metric:

1. Log on to the device using a valid user name and password for read-write access.
2. Click on the plus sign next to Configure in the tree view to expand the list of configuration options.
3. Click on the plus sign next to DVMRP in the tree view to expand the list of DVMRP option links.
4. Select the Virtual Interface link to display a table listing the configured DVMRP Interfaces.
5. Click on the Modify button next to the interface you want to modify. The DVMRP Interface configuration panel is displayed.
6. Enter a value from 1 – 31 in the Metric field.

7. Click the Add button to save the changes to the device's running-config file.
8. 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.

NOTE: You also can access the dialog for saving configuration changes by clicking on Command in the tree view, then clicking on Save to Flash.

Enabling Advertising

You can turn the advertisement of a local route on (enable) or off (disable) on the interface. By default, advertising is enabled.

USING THE CLI

To enable advertising on an interface, enter the following:

```
BigIron(config-if-1/4)# ip dvmrp advertise-local on
```

Syntax: advertise-local on | off

USING THE WEB MANAGEMENT INTERFACE

To enable local advertising on a DVMRP interface:

1. Log on to the device using a valid user name and password for read-write access.
2. Click on the plus sign next to Configure in the tree view to expand the list of configuration options.
3. Click on the plus sign next to DVMRP in the tree view to expand the list of DVMRP option links.
4. Select the Virtual Interface link to display a table listing the configured DVMRP Interfaces.
5. Click on the Modify button next to the interface you want to modify. The DVMRP Interface configuration panel is displayed.
6. Select Enable next to Advertise Local.
7. Click the Add button to save the changes to the device's running-config file.
8. 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.

Displaying Information About an Upstream Neighbor Device

In software release 07.7.00 and later, 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.

```
BigIron# show ip dvmrp rpf 1.1.20.2
directly connected or via an L2 neighbor
BigIron# show ip dvmrp rpf 1.2.3.4
no route
BigIron# 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. For more information about this command, see the *Foundry Switch and Router Command Line Interface Reference*.

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.

NOTE: IP tunnels are supported for DVMRP only in software release 07.6.01 and later.

EXAMPLE:

To configure an IP tunnel as seen in Figure 14.9, enter the IP tunnel destination address on an interface of the router.

USING THE CLI

To configure an IP address on Router A, enter the following:

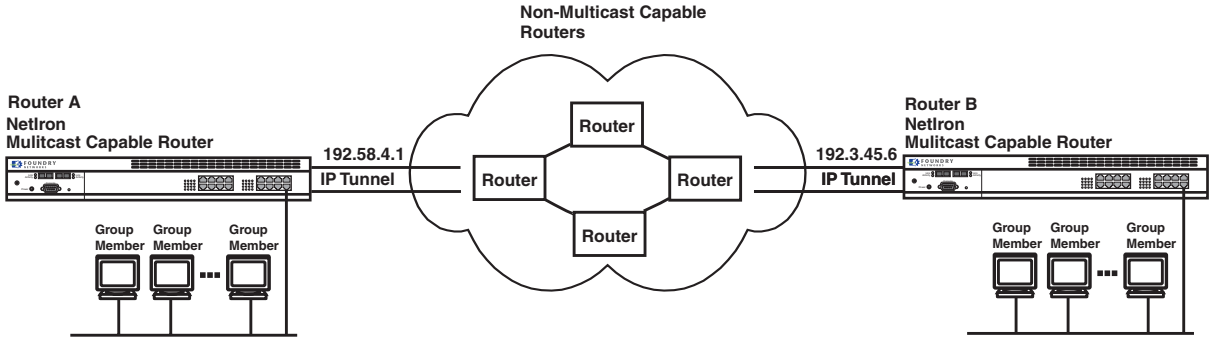
```
NetIron(config)# int e1
NetIron(config-if-1)# ip tunnel 192.3.45.6
```

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:

```
NetIron(config-if-1)# ip tunnel 192.58.4.1
```

Figure 14.9 IP in IP tunneling on multicast packets in a unicast network



USING THE WEB MANAGEMENT INTERFACE

1. Log on to the device using a valid user name and password for read-write access.

2. Click on the plus sign next to Configure in the tree view to expand the list of configuration options.
3. Click on the plus sign next to PIM in the tree view to expand the list of PIM option links.
4. Click on the Virtual Interface link to display the PIM Interface configuration panel.

NOTE: If the device already has PIM interfaces, a table listing the interfaces is displayed. Click the Modify button to the right of the row describing an interface to change its configuration, or click the Add Virtual Interface link to display the PIM Interface configuration panel.

---

5. Select the interface type. You can select Subnet or Tunnel. In this case, select Tunnel.
6. Select the IP address of the interface being configured from the Local Address pulldown menu.
7. Enter the IP address of the destination interface, the end point of the IP Tunnel, in the Remote Address field. IP tunneling must also be enabled and defined on the destination router interface as well.
8. Modify the time to live threshold (TTL) if necessary. The TTL defines the minimum value required in a packet in order for the packet to be forwarded out the interface.

NOTE: 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 values are 1 – 64. The default value is 1.

---

9. Click Enable or Disable next to Advertise Local to enable or disable the feature.
10. Click Enable or Disable next to Encapsulation to enable or disable the feature.
11. Click the Add button to save the change to the device's running-config file.
12. 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.
13. Repeat the steps above on the router that has the interface on the remote end of the IP tunnel.

Using ACLs to Control Multicast Features

Starting with Release 07.6.03, 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

Starting with software release 07.6.03, 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:

```
BigIron(config)# access-list 2 permit 239.255.162.0 0.0.0.255

BigIron(config)# router pim
BigIron(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:

```
BigIron(config)# access-list 5 deny host 239.255.162.2

BigIron(config)# access-list 5 permit 239.255.0.0 0.0.255.255

BigIron(config)# router pim
```



```
BigIron(config-pim-router)# bsr-candidate ve 43 32 100
BigIron(config-pim-router)# rp-candidate ve 43
BigIron(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:

```
BigIron(config)# access-list 44 permit 239.255.162.0 0.0.0.255
```

```
BigIron(config)# router pim
BigIron(config-pim-router)# rp-address 43.43.43.1
BigIron(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,

```
BigIron(config) #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.

```
BigIron(config)# 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:

```
BigIron(config)# interface ethernet 1/1
BigIron(config-if-1/1)# ip address 99.99.99.5 255.255.255.0
BigIron(config-if-1/1)# ip pim-sparse
BigIron(config-if-1/1)# exit

BigIron(config)# access-list 5 deny host 239.255.162.2
BigIron(config)# access-list 5 permit 239.0.0.0 0.0.255.255

BigIron(config)# router pim
BigIron(config-pim-router)# bsr-candidate ethernet 1/1 32 100
BigIron(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:

```
BigIron(config)# interface ve 16
BigIron(config-vif-16)# ip address 16.16.16.1 255.255.255.0
BigIron(config-vif-16)# ip pim-sparse
BigIron(config-vif-16)# exit

BigIron(config)# access-list 5 deny host 239.255.162.2
BigIron(config)# access-list 5 permit 239.255.0.0 0.0.255.255

BigIron(config)# router pim
BigIron(config-pim-router)# bsr-candidate ve 16 32 100
BigIron(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:

```
BigIron(config)# interface loopback 1
BigIron(config-lbif-1)# ip address 88.88.88.8 255.255.255.0
BigIron(config-lbif-1)# ip pim-sparse
BigIron(config-lbif-1)# exit
```



```
BigIron(config)# access-list 5 deny host 239.255.162.2
BigIron(config)# access-list 5 permit 239.255.0.0 0.0.255.255

BigIron(config)# router pim
BigIron(config-pim-router)# bsr-candidate loopback 1 32 100
BigIron(config-pim-router)# rp-candidate loopback 1 group-list 5
```

Syntax: [no] rp-candidate ethernet <portnum> | loopback <num> | ve <num> [group-list <access-list-num>]

The **ethernet** <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** <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.

Using ACLs to Control Multicast Traffic Boundaries

You can create ACLs that determine which multicast traffic packets can be forwarded on an interface in a PIM or DVMRP domain. The ACLs can be create to be applied to a range of multicast group addresses. If an ACL denies the specified multicast group addresses, incoming or outgoing packets from those addresses will not be allowed to flow across the interface.

For example, to set up a boundary, which will deny all multicast group addresses within the 239.x.x.x IP address range, enter commands such as the following:

```
BigIron(config)# access-list 1 deny 239.0.0.0 0.255.255.255
BigIron (config)# access-list 1 permit 234.0.0.0 0.255.255.255

BigIron(config)# interface ethernet 1/1
BigIron(config-if-1/1)# ip pim-sparse
BigIron(config-if-1/1)# ip multicast boundary 1
```

Syntax: [no] ip multicast boundary <access-list-num>

The <access-list-num> parameter defines the ACLs used to set-up the boundaries for multicast traffic packets.

NOTE: Extended ACLs cannot be used in this feature. Also, this feature is not supported on the NetIron 40G.

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 a static route for a multicast source network, use one of the following methods.

USING THE CLI

To add static routes to multicast router A (see Figure 14.10), 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 <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** <portnum> parameter to specify a physical port or the **ve** <num> parameter to specify a virtual interface.

NOTE: The **ethernet** <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.

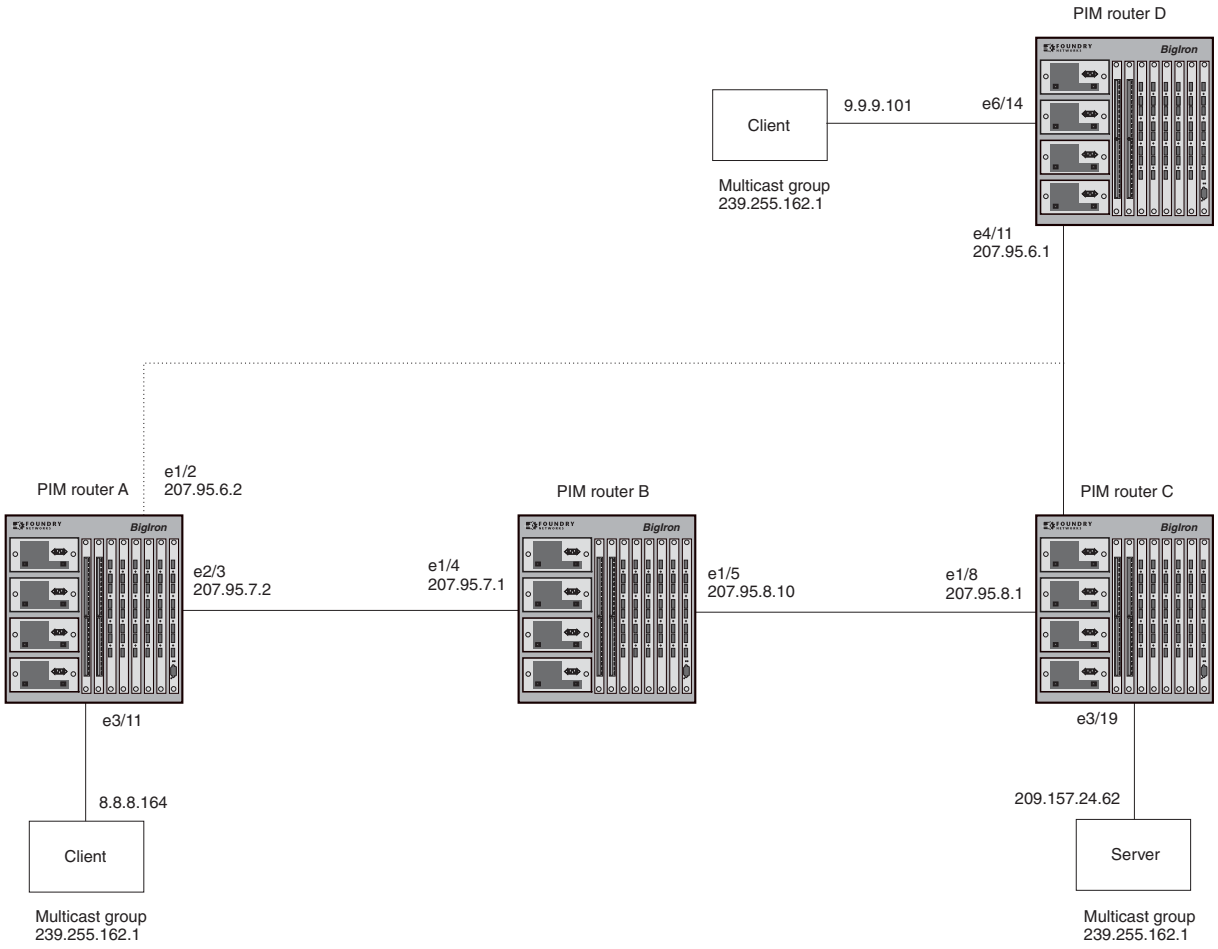
The **rpf_address** <rpf-num> parameter specifies an RPF number.

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 14.10 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 14.10 Example multicast static routes



To add a static route to a virtual interface, enter commands such as the following:

```
BigIron(config)# mroute 3 0.0.0.0 0.0.0.0 int ve 1 distance 1
BigIron(config)# write memory
```

USING THE WEB MANAGEMENT INTERFACE

You cannot configure a static multicast route using the Web management interface.

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.

USING THE CLI

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:

```
BigIron# 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>

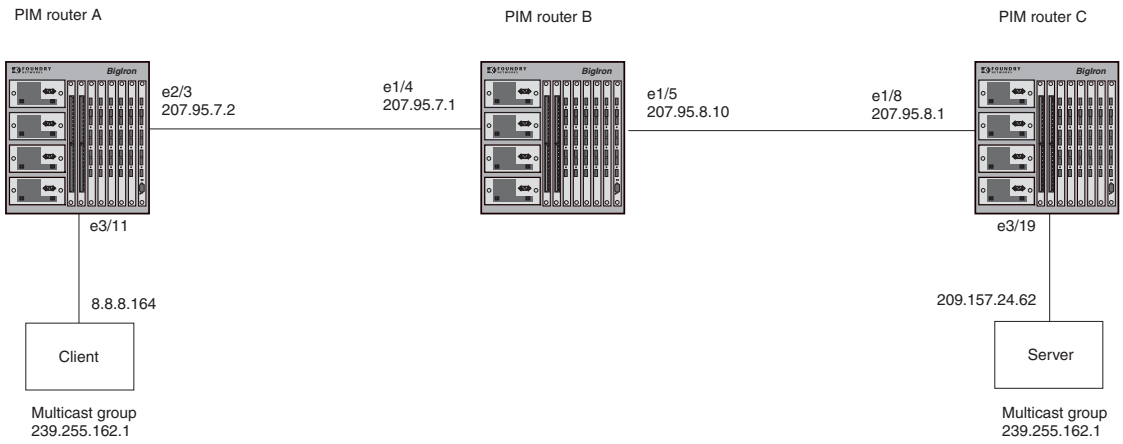
The **source** <ip-addr> parameter specifies the address of the route's source.

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 14.11 shows an example of an IP multicast group. The command example shown above is entered on PIM router A.

Figure 14.11 Example PIM Group



The command example above indicates that the source address 209.157.24.62 is three hops (three PIM routers) away from PIM router A. In PIM terms, each of the three routers 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 router hops between the source and destination.

USING THE WEB MANAGEMENT INTERFACE

You cannot trace a PIM route using the Web management interface.

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.

USING THE CLI

To display another PIM router's PIM configuration, enter a command such as the following:

```
BigIron# 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 14.11 on page 14-84. 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.

USING THE WEB MANAGEMENT INTERFACE

You cannot display another router's PIM configuration using the Web management interface.

IGMP V3

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 07.8.00; 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:

```
BigIron(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:

```
BigIron(config)# interface eth 1/5
BigIron(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:

```
BigIron(config)# interface ve 3
BigIron(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:

```
BigIron(config)# interface ve 3
BigIron(config-vif-3)# ip igmp version 2
BigIron(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 <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.

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:

```
BigIron(config)# interface ve 13
BigIron(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:

```
BigIron(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, on BigIron Chassis devices, 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. Refer to the *Foundry Enterprise Configuration and Management Guide* for details.

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:

```
BigIron(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, on BigIron Chassis devices, 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. Refer to the *Foundry Enterprise Configuration and Management Guide* for details.

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:

```
BigIron(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

USING THE CLI

To enable the SSM protocol on a Foundry device running PIM-SM, enter a command such as the following:

```
BigIron(config)# router pim
BigIron(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.

USING THE WEB MANAGEMENT INTERFACE

You cannot change this parameter using the Web management interface.

Displaying IGMP V3 Information

The sections below present the show commands available for IGMP V3.

Displaying IGMP Group Status

NOTE: This report is available on Layer 3 Switches.

You can display the status of all IGMP multicast groups on a device by entering the following command:

```
BigIron(config)# 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:

```
BigIron(config)# 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:

```
BigIron(config)# 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.

IGMP V2 and V3 statistics displayed on the report for each interface.

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.
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.

This Field	Displays
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

NOTE: This report is available on Layer 3 Switches.

You can display the status of a multicast enabled port by entering a command such as the following:

```
BigIron(config)# 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:

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

NOTE: This report is available on Layer 3 Switches.

To display the traffic status on each virtual routing interface, enter the following command:

```
BigIron(config)# 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:

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:

```
BigIron# clear igmp traffic
```

Syntax: clear igmp traffic

This command clears all the multicast traffic information on all interfaces on the device.

IGMP V3 Snooping

Release 08.0.00 supports IGMP V3 snooping on devices running both switch and router code. This section consists of the following topics:

IGMP V3 Snooping Overview

This section presents an overview of IGMP V3 snooping on Foundry devices. For additional information on IGMP V3, see the “IGMP V3” on page 14-86.

NOTE: IGMP V3 Snooping is supported on devices running Enterprise software release 08.0.00 and later.

Using IGMP Protocols on Foundry Devices

The default behavior for a Foundry device to handle multicast packets is to broadcast them to every port in the VLAN, except the incoming port (unless the **route-only** command is configured). Internet Group Management Protocol (IGMP) protocols allow the Foundry device to forward multicast traffic to the ports that want it, and stop forwarding multicast traffic to ports that don't want it.

Clients (hosts) send IGMP membership reports to an IGMP-enabled Foundry device to indicate the requested traffic stream. The Foundry device periodically broadcasts IGMP queries to all ports in the VLAN. A client must send membership reports immediately when it first intends to receive multicast traffic. The client also responds to queries from a Foundry device, and finally sends out a leave message when it no longer wants traffic.

An IGMP-enabled Foundry device's responsibility is to broadcast general queries periodically, and to send group queries when receiving a leave message to make sure that no other clients on the same port still want this specific traffic before removing traffic from the port.

IGMP V2 lets clients specify which group (destination IP address) to receive. The protocol cannot choose the source of the traffic. In contrast, IGMP V3 is for source-specific multicast traffic. It adds the capability for clients to include or exclude specific traffic sources. An IGMP V3 device's port state could be in include or exclude mode; there are six types of group records for client reports. See "IGMP V3" in the *Foundry Enterprise Configuration and Management Guide* for detailed IGMP V3 information.

IGMP protocols provide a scheme for clients and a router to exchange messages and let the device build a database indicating which port wants what traffic. IGMP protocols do not specify forwarding methods, however. IGMP snooping or multicast protocols such as PIM or DVMRP are required 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. Servers (traffic sources) are not required to send IGMP memberships. For basic IGMP snooping information, see the "Configuring IP Multicast Traffic Reduction" chapter in the *Foundry Switch and Router Installation and Basic Configuration Guide*.

IGMP Snooping Support on Foundry Devices

IGMP V2 snooping has been supported on Foundry devices running switch software from release 04.x.x. Foundry devices running router software support IGMP V2 snooping using Layer 4 CAM starting with release 07.7.00, as well as IGMP V3 for PIM and DVMRP starting with release 07.8.00. Release 08.0.00 supports IGMP V3 snooping on both router and switch software images.

IGMP V3 is a source-specific protocol, and requires Layer 4 CAM to match both source and group. In 08.0.00 switch software, you can specify whether the Foundry device uses Layer 2 or Layer 4 CAM for IGMP snooping. For backward compatibility, the default is to use Layer 2 CAM. You can optionally configure the system to use Layer 4 CAM; otherwise Foundry devices running switch software use Layer 2 CAM. If you configure IGMP V3 on a VLAN, the VLAN uses Layer 4 CAM. In this case, a switch could have some VLANs using the default Layer 2 CAM, and others using Layer 4 CAM due to IGMP V3 configuration.

When Layer 2 CAM is used, traffic is switched solely based on 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. As a result, the use of Layer 4 CAM should be avoided if there are many (for example, one thousand) different source and group pairs.

Configuring Queriers and Non-Queriers

An IGMP-enabled Foundry device can be configured to be 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 release 08.0.00, VLANs can be independently configured to be queriers or non-queriers. If a VLAN has a connection to a PIM or DVMRP-enabled port on another router, the VLAN should be configured as a non-querier. When two IGMP snooping devices are connected together, and there is no connection to a PIM or DVMRP-enabled port, then one of the devices should be configured as a querier. If both devices are configured as queriers, then one of them will stop sending queries after the two devices exchange queries. If the querier is a

switch, it must have a global IP address in order to send queries. If the querier is a router, the snooping VLAN's router interface must have an IP address, or the router must have a global loopback address.

VLAN-Specific Configuration

In this release, you can configure snooping on some VLANs or all VLANs. Each VLAN can independently enable or disable IGMP or PIM snooping, or can be configured with IGMP V2 or 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 supercede the global **ip multicast...** commands. Per-VLAN configuration is available starting in release 08.0.00.

The configuration of IGMP for snooping and for PIM/DVMRP are independent. The **ip igmp...** commands, available in router code, set IGMP parameters used by PIM/DVMRP, while the **ip multicast...** and **multicast...** commands apply to snooping. In router code, if snooping is configured for a VLAN, and the VLAN's associated virtual interface has PIM or DVMRP enabled, then PIM or DVMRP has higher priority over snooping. The output of the **show multicast vlan** command indicates whether snooping is disabled on a VLAN because PIM or DVMRP is enabled.

Using IGMP V2 with IGMP V3

As with the IGMP V3 functionality introduced in release 07.8.00, snooping can be configured as IGMP V2 or V3 on individual ports on a VLAN.

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 be able to process them. For example, a Foundry device running switch code using Layer 2 CAM can recognize IGMP V3 packets, but does not have the source-specific switching capability. In this case, the switch forwards traffic of the group, but does not consider the source information.

Also, a device running IGMP V3 can recognize and process IGMP V2 packets, but when that device 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. The **show ip multicast traffic** command displays the number of packets received on a port that have a different IGMP version.

Support for IGMP V2 Snooping in Layer 3 Software Images on FES

IGMP V2 snooping has been supported on FES devices running *switch* software since release 01.0.00. Release 03.4.00 supports IGMP V2 snooping on both *router* and *switch* software images. On both images, IGMP V2 snooping is MAC-based. This differs from IGMP V2 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 FES router images match on Layer 2 destination MAC address entries.

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.

NOTE: 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.

Configuring IGMP V3 Snooping

Configuring IGMP V3 snooping on a Foundry device consists of the following global and VLAN-specific tasks:

Global tasks:

- Configuring the IGMP mode: active or passive
- Modifying the age interval

- Configuring filtering for multicast groups
- Dropping IGMP V3 traffic in hardware
- Specifying the interval for query messages (active IGMP mode only)
- Specifying that all VLANs use Layer 4 CAM for IGMP snooping
- Specifying the global IGMP version

VLAN specific tasks:

- Configuring the IGMP mode for the VLAN: active or passive
- Enabling or disabling IGMP snooping for the VLAN
- Enabling or disabling PIM SM snooping
- Configuring the IGMP version for the VLAN
- Configuring the IGMP version for individual ports in the VLAN
- Enabling client tracking and the fast-leave feature

Configuring the Global IGMP Mode

You can use active or passive IGMP mode on the Foundry device. The default mode is passive. If you specify an IGMP version 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 IP multicast groups on the network and makes entries in the IGMP table based on the Group Membership reports received from the network.
- Passive – When passive IGMP mode is enabled, the device listens for IGMP Group Membership reports but does not send IGMP queries.

To set the IGMP mode for the Foundry device to active, enter the following command:

```
BigIron(config)# ip multicast active
```

Syntax: [no] ip multicast [active | passive]

If you omit both the **active** and **passive** keywords, it is equivalent to entering **ip multicast passive**.

Modifying the Age Interval

When the Foundry device receives a Group Membership report, it makes an entry in the IGMP 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.

To modify the age interval, enter a command such as the following:

```
BigIron(config)# ip multicast age-interval 280
```

Syntax: [no] ip multicast age-interval <interval>

The <interval> parameter specifies the interval between queries. You can specify a value from 10 – 1220 seconds. The default is 140 seconds.

Configuring Filtering for Multicast Groups

By default, Foundry devices forward multicast traffic for all valid multicast groups. You can configure a Foundry device to filter out all multicast traffic for groups other than the ones for which the device has received Group Membership reports.

When device starts up, it forwards all multicast groups even though IGMP snooping is configured. This process continues until the device receives a group membership report. Once the group membership report is received, the device drops all multicast packets for groups other than the ones for which the device has received the group membership report. When there are no members in a group, the device drops all multicast packets for that group. If you have multicast applications for which the client never sends a group membership report, you should not enable IP multicast filtering.

To enable IP multicast filtering, enter the following command:

```
BigIron(config)# ip multicast filter
```

Syntax: [no] ip multicast filter

Dropping IGMP V3 Traffic in Hardware

When there are no clients for a flow, and the **ip multicast filter** command is configured, you can configure the device to drop the IGMP V3 traffic in hardware. Note that this feature does not apply to VLANs using Layer 2 CAM.

To cause IGMP V3 traffic to be dropped in hardware when there are no clients for a flow, enter the following command:

```
BigIron(config)# ip multicast hardware-drop
```

Syntax: [no] ip multicast hardware-drop

Modifying the Query Interval (Active IGMP Mode Only)

If the IGMP mode is set to active, you can modify the query interval, which specifies how often a Foundry device sends Group Membership queries.

To modify the query interval, enter a command such as the following:

```
BigIron(config)# ip multicast query-interval 120
```

Syntax: [no] ip multicast query-interval <interval>

The <interval> parameter specifies the interval between queries. You can specify a value from 10 – 600 seconds. The default is 60 seconds.

Specifying that all VLANs Use Layer 4 CAM for IGMP Snooping (Switch Code Only)

On Foundry devices running switch code, the default for IGMP snooping is to use Layer 2 CAM, which matches on destination MAC address. Since IGMP V3 is a source-specific protocol, it requires Layer 4 CAM.

When you configure IGMP V3 on a VLAN, Layer 4 CAM is used for that VLAN. If IGMP V2 snooping is configured on a VLAN, Layer 2 CAM is used for that VLAN. Thus, some VLANs can use Layer 2 CAM, while others use Layer 4 CAM. You can optionally configure the device so that Layer 4 CAM is used for all VLANs.

To do this, enter the following command:

```
BigIron(config)# ip multicast use-l4-cam
```

Syntax: [no] ip multicast use-l4-cam

This command applies to Foundry devices running switch code only. On Foundry devices running router code, Layer 4 CAM is used for all VLANs.

Configuring the Global IGMP Version

You can specify the global IGMP version used on the Foundry device, either IGMP V2 or IGMP V3. For example, the following command causes the Foundry device to use IGMP V3:

```
BigIron(config)# ip multicast version 3
```

Syntax: [no] ip multicast version 2 | 3

In addition, you can optionally specify the IGMP version for individual VLANs, or individual ports within VLANs. If no IGMP version is specified for a VLAN, then the globally configured IGMP version is used. If an IGMP 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.

Configuring the IGMP Mode for a VLAN

You can use active or passive IGMP mode on a VLAN. The default mode is passive. The setting specified for the VLAN overrides the global setting.

- Active – When active IGMP mode is enabled, a Foundry device actively sends out IGMP queries to identify IP

multicast groups on the network and makes entries in the IGMP table based on the Group Membership reports received from the network.

- Passive – When passive IGMP mode is enabled, the device listens for IGMP Group Membership reports but does not send IGMP queries.

To set the IGMP mode for VLAN 20 to active, enter the following commands:

```
BigIron(config)# vlan 20
BigIron(config-vlan-20)# multicast active
```

Syntax: [no] multicast active | passive

Disabling IGMP Snooping for the VLAN

When IGMP snooping is enabled globally, you can 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.

```
BigIron(config)# vlan 20
BigIron(config-vlan-20)# disable-igmp-snoop
```

Syntax: [no] disable-igmp-snoop

Disabling PIM SM Snooping for the VLAN

When PIM SM snooping is enabled globally, you can disable it for a specific VLAN. For example, the following commands cause PIM SM snooping to be disabled for VLAN 20. This setting overrides the global setting.

```
BigIron(config)# vlan 20
BigIron(config-vlan-20)# disable-pimsm-snoop
```

Syntax: [no] disable-pimsm-snoop

Configuring the IGMP Version for the VLAN

You can specify the IGMP version for the ports in a VLAN. For example, the following commands cause VLAN 20 to use IGMP V3:

```
BigIron(config)# vlan 20
BigIron(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 version specified for the VLAN.

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 cause ports 4/3, 4/5, 4/6, and 4/7 to use IGMP V3. The other ports in the VLAN use the IGMP version specified with the **multicast version** command, or if the **multicast version** command is not configured, the globally configured IGMP version is used.

```
BigIron(config)# vlan 20
BigIron(config-vlan-20)# multicast port-version 3 ethe 4/3 ethe 4/5 to 4/7
```

Syntax: [no] multicast port-version 2 | 3 <port-numbers>

Enabling Membership Tracking and Fast Leave for the VLAN

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.

When the tracking feature is enabled, the device immediately stops forwarding multicast traffic to the interface (without waiting three seconds) if an IGMP V3 client sends a leave message and there are no other 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.

To enable the tracking and fast leave feature for VLAN 20, enter the following commands:

```
BigIron(config)# vlan 20
BigIron(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.

Displaying IGMP V3 Snooping Information

You can display the following IGMP V3 snooping information:

- IGMP error information
- Information about VLANs using Layer 4 CAM
- Hardware resource usage for VLANs using Layer 2 CAM
- Group and forwarding information for VLANs using Layer 2 CAM
- Multicast forwarding cache information
- PIM SM snooping information for VLANs using Layer 2 CAM
- IGMP V3 memory pool usage
- Status of IGMP V3 traffic
- IGMP V3 information by VLAN

Displaying IGMP Error Information

To display information about possible IGMP errors, enter the following command:

```
BigIron# show ip multicast error
**** Warning! counter igmp checksum error = 10
**** Warning! counter igmp, pkt buf alloc fail = 7
**** Warning! counter snoop router fid alloc fail = 12
```

Syntax: show ip multicast error

Displaying Information about VLANs Using Layer 4 CAM

You can display the status of all or specific groups of VLANs using Layer 4 CAM by entering the following command:

```
BigIron# show ip multicast group
VL20 : 1 groups, 1 group-port
      group           phy-port static querier life mode    #_src
1     224.1.1.1       e4/12    no     no      140  exclude 0
```

To display detailed information, enter the following command:

```
BigIron# show ip multicast group 224.1.1.1 detail
Display group 224.1.1.1 in all interfaces in details.
VL20 : 1 groups, 1 group-port
      group           phy-port static querier life mode    #_src
1     224.1.1.1       e4/12    no     no           include 2
    group: 224.1.1.1, include, permit 2 (source, life):
      (1.1.32.1 120) (1.1.32.2 120)
```

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:

```
BigIron# show ip multicast group 224.1.1.1 track
Display group 224.1.1.1 in all interfaces with tracking enabled.
VL20 : 1 groups, 1 group-port, tracking_enabled
      group           phy-port static querier life mode    #_src
1     224.1.1.1       e4/12    no     no           include 2
    receive reports from 1 clients:
      2.2.2.100
```

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>. Omit the <group-address> if you want a report for all multicast groups using Layer 4 CAM.

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.

Table 14.5 describes the information displayed by the **show ip multicast group** command.

Table 14.5: Output from the show ip multicast group command

This Field	Displays
Group	The address of the multicast group (destination IP address)
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. Since static groups can be configured only for PIM/DVMRP, this should always be "no" for snooping.
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.

Table 14.5: Output from the show ip multicast group command

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.
#_src	Identifies the source list that will be included or excluded on the interface. If an 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 Hardware Resource Usage for VLANs Using Layer 2 CAM

To display information about the hardware resources used by VLANs using Layer 2 CAM, enter the following command. To display hardware resource usage information about VLANs using Layer 4 CAM, enter the **show ip multicast mcache** command.

NOTE: This command is available on Foundry devices running switch code only.

```
FastIron# show ip multicast hardware
This displays L2 cam, use "sh ip mu mcache" for L4 cam
Hw resource is shared by groups with the same lower 23 bits
VLAN ID 20
Total number of HW resource in vlan: 1
1     Group: 0.1.1.1, fid 08a7, cam 2056
   Forwarding Port: 4/2 4/7 4/12
```

Syntax: show ip multicast hardware

Displaying Group and Forwarding Information for VLANs Using Layer 2 CAM

To display group and forwarding information for VLANs using Layer 2 CAM, enter the following command.

```
FastIron# show ip multicast l2-group
IP multicast is globally enabled - Passive
VL20: dft V2, L2 CAM, glb cfg Passive, pimsm (glb cfg), 1 grp
   Querier: 1.1.20.1, (port: 4/1)
   Router fid=000008A6, Ports:  4/1 4/3
   Total number of Multicast Group in vlan: 1
1     Group: 224.1.1.1, fid 08a7, cam 2057
   Forwarding Port: 4/2 4/7 4/12
```

Syntax: show ip multicast l2-group

Table 14.6 describes the information displayed by the **show ip multicast l2-group** command.

Table 14.6: Output from the show ip multicast l2-group command

This Field...	Displays...
IP multicast state	The first line of the display indicates whether IGMP V3 snooping is enabled or disabled.
VLAN ID	The port-based VLAN to which the information listed below the VLAN ID applies. Each port-based VLAN is a separate Layer 2 broadcast domain. The “glb cfg passive” in the display is due to the global **ip multicast passive** statement in the configuration.
Querier	The querier’s IP address
Ports	The ports that are connected to routers that support IP multicast. Router ports are the ports that receive queries.
Total Number of Multicast Group in vlan	The number of groups for which the VLAN’s ports have received IGMP group membership reports, join messages, or prune messages.
Group	An IP multicast group.
Forwarding Port	The ports to which the traffic is sent.

To clear group and forwarding information for VLANs using Layer 2 CAM, enter the following command:

```
FastIron# clear ip multicast all
```

Syntax: clear ip multicast all

To clear group and forwarding information for a specific group of VLANs that uses Layer 2 CAM, enter a command such as the following:

```
FastIron# clear ip multicast group 224.1.1.1
```

Syntax: clear ip multicast group <group>

The **clear ip multicast all** and **clear ip multicast group** commands are available on Foundry devices running switch code only.

Displaying Multicast Forwarding Cache Information

The multicast forwarding cache contains multicast information for VLANs using Layer 4 CAM. To display information in the multicast forwarding cache, enter the following command:

```
FastIron# show ip multicast mcache
mcache is only available for vlans using L4 cam
Example: (S G) in tag e4/3 cnt=: e4/3 is input, cnt: SW proc. count   <--- new
    OIF: TR(e4/1, e4/2) (20): 4/1 is primary trunk, 4/2 is real out, (20):
age  <--- new
vlan 1, has 0 cache
vlan 20, has 3 cache
1    (1.1.32.3 224.1.1.1) in tag e4/3, cnt=1
     OIF: tag TR(e4/1,e4/1) (20), e4/7 (20),
     age=0s up-time=0m fid=2d7f, cam=00aca,
2    (1.1.32.2 224.1.1.1) in tag e4/3, cnt=1
     OIF: tag TR(e4/1,e4/2) (20), e4/7 (20),
     age=0s up-time=0m fid=2d43, cam=00be1,
3    (1.1.32.0 224.1.1.1) in tag e4/3, cnt=2
     OIF: tag TR(e4/1,e4/1) (20), e4/7 (20),
     age=0s up-time=0m fid=08b5, cam=00aa0,
```

Syntax: show ip multicast mcache

Table 14.7 describes the information displayed by the **show ip multicast mcache** command.

Table 14.7: Output from the show ip multicast mcache command

This Field...	Displays...
in	The traffic input port.
tag	Whether the port is tagged.
cnt	The number of packets processed in software. This does not include hardware-switched packets.
OIF	The output interface.
TR(e4/1, e4/2)	Trunk port information. In this example, e4/1 is the primary port, e4/2 is the real output port.
(20)	Amount of time since receiving IGMP membership for this port.

To clear the multicast forwarding cache of information about VLANs using Layer 4 CAM, enter the following command:

```
FastIron# clear ip multicast mcache
```

Syntax: clear ip multicast mcache

To clear the multicast forwarding cache of information about a specific VLAN that uses Layer 4 CAM, enter a command such as the following:

```
FastIron# clear ip multicast vlan 20
```

Syntax: clear ip multicast vlan <vlan-id>

Displaying PIM Sparse Snooping Information for VLANs Using Layer 2 CAM

To display PIM sparse snooping information for VLANs using Layer 2 CAM, use the **show ip multicast pimsm** command. This command is available on devices running switch code only.

For example:

```
FastIron# show ip multicast pimsm
Display vlan using L2 CAM. Use "show ip multicast mcache" for vlan using L4 CAM
Use "sh pimsm-snooping A.B.C.D" to show sources of a specific group
VLAN ID 20, total 1 entries
PIMSM Neighbor list:
        1.1.20.2         : e4/3 (primary-trunk 4/3), expire 180 s
        1.1.20.1         : e4/1 (primary-trunk 4/1), expire 150 s
        1.1.20.4         : e4/7 expire 140 s
1     Group: 224.1.1.1, fid 08a7, cam 2056
   Forwarding Port: 4/1 4/7 4/12
   PIMv2 Group Port: e4/1 e4/7
   (Source, Port) list: (age), 4 entries
```

Syntax: show ip multicast pimsm

When a VLAN uses Layer 4 CAM, all PIM sparse join and prune messages are directly added to the multicast forwarding cache; to display PIM sparse snooping information for VLANs using Layer 4 CAM, use the **show ip multicast mcache** command.

Displaying IGMP V3 Memory Pool Usage

You can display information about the memory pools used for IGMP. Foundry devices running router code have a single set of memory pools, and devices running switch code have a separate set of memory pools for Layer 2 CAM and Layer 4 CAM.

For example, the following output is from a device running switch software. Memory pool information for VLANs using Layer 2 CAM is displayed, followed by memory pool information for VLANs using Layer 4 CAM.

```
FastIron# show ip multicast resource
Resource used by vlans using L2-cam
                     alloc in-use  avail allo-fail up-limit   get-mem

pim neighbor             32      0     32         0      512         0
.... other entries removed
igmp leave              512      0    512         0 no-limit         0
In use: hw-res: 0, cam: 0, fid: 0

Resource used by vlans using L4-cam
                     alloc in-use  avail allo-fail up-limit   get-mem
igmp group               512      0    512         0    32000       367
.... other entries removed
snoop flow map         10240      3  10237         0    10240     96463
total pool memory 293824 bytes
```

Syntax: show ip multicast resource

Displaying Status of IGMP V3 Traffic

To display status information for IGMP V3 traffic, enter the following command:

```
FastIron# show ip multicast traffic
  Total Recv: 42829, Xmit: 0 (including IGMP for PIM/DVMRP)
Recv  QryV2 QryV3 G-Qry GSQry MbrV2 MbrV3 Leave  IsIN  IsEX ToIN ToEX ALLO  BLK
VL1       0     0     0     0     0     0     0     0     0    0    0    0    0
VL20    193 34551     1     3  7704    12   365     9  7704  365    0    0    3
Send  QryV1 QryV2 QryV3 G-Qry GSQry
VL1       0     0     0     0     0
VL20      0     0     0     0     0
VL1    pimsm-snooping, Hello:     0,  Join/Prune:     0
VL20   pimsm-snooping, Hello: 44683,  Join/Prune: 32849
```

Syntax: show ip multicast traffic

Table 14.8 describes the information displayed by the **show ip multicast traffic** command.

Table 14.8: Output from the show ip multicast traffic command

This Field	Displays
QryV2	Number of general IGMP V2 queries received or sent by the virtual routing interface.
QryV3	Number of general IGMP V3 queries received or sent by the virtual routing interface.
G-Qry	Number of group specific queries received or sent by the virtual routing interface.
GSQry	Number of source specific queries 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.
ALLO	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.

To clear the counters for IGMP V3 traffic, enter the following command:

```
FastIron# clear ip multicast traffic
```

Syntax: clear ip multicast traffic

Displaying IGMP V3 Information by VLAN

You can display IGMP V3 information for all VLANs or for a specific VLAN. For example, to display multicast information for VLAN 20, enter the following command:

```
FastIron# show ip multicast vlan 20
version=2, query-t=20, group-aging-t=140, max-resp-t=3
VL20: dft V2, L4 CAM, glb cfg Passive, pimsm (glb cfg), 0 grp, 4 caches, , rtr-
fid=08A6
    router ports: e4/3(160), e4/7(180), e4/1(140),
  e4/7    has    0 groups, non-QR (passive), default V2
  e4/3    has    0 groups, non-QR (passive), default V2
  e4/1    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.

Table 14.9 describes information displayed by the **show ip multicast vlan** command.

Table 14.9: Output from 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 multicast groups can be members of this group before aging out.
rtr-fid	The FID of the ports receiving queries
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.

Chapter 15
Configuring OSPF

This chapter describes how to configure OSPF on Foundry Layer 3 Switches using the CLI and Web management interface.

To display OSPF configuration information and statistics, see “Displaying OSPF Information” on page 15-49.

For complete syntax information for the CLI commands shown in this chapter, see the *Foundry Switch and Router Command Line Interface Reference*.

NOTE: The TurboIron/8, Stackable NetIron, and Chassis Layer 3 Switches using basic management modules (not Management 2 or higher) can contain 10000 routes by default. If you need to increase the capacity of the IP route table, see the “Displaying and Modifying System Parameter Default Settings” section of the “Configuring Basic Features” chapter of the *Foundry Switch and Router Installation and Basic Configuration Guide*.

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.

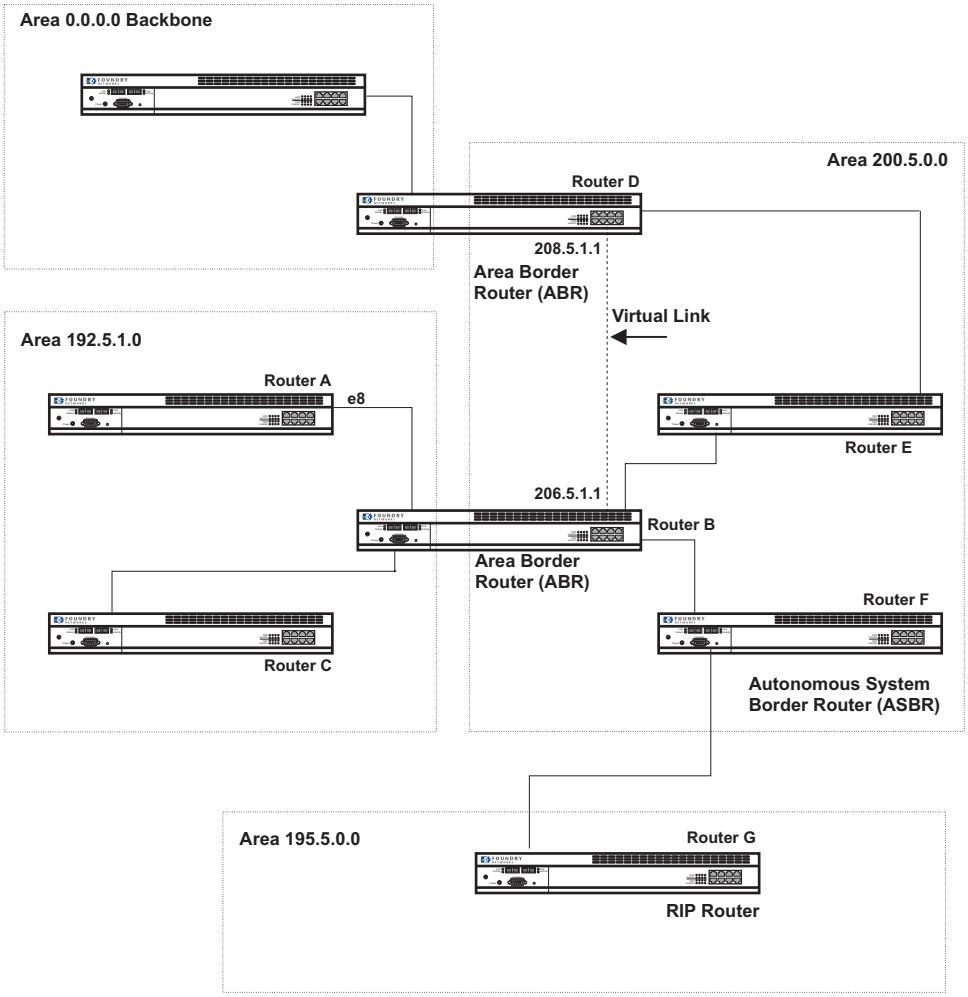
An AS can be divided into multiple ***areas*** as shown in Figure 15.1 on page 15-2. Each area represents a collection of contiguous networks and hosts. Areas limit the area to which link-state advertisements are broadcast, thereby limiting the amount of flooding that occurs within the network. An area 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 15-35.

Figure 15.1 OSPF operating in a network



OSPF Point-to-Point Links

OSPF point-to-point links are supported in the following software releases:

- Enterprise software release 07.8.00 and later on 10/100 and Gigabit Ethernet interfaces
- BigIron MG8 and NetIron 40G, this command is available in software release 02.2.01 and later

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 15-47.

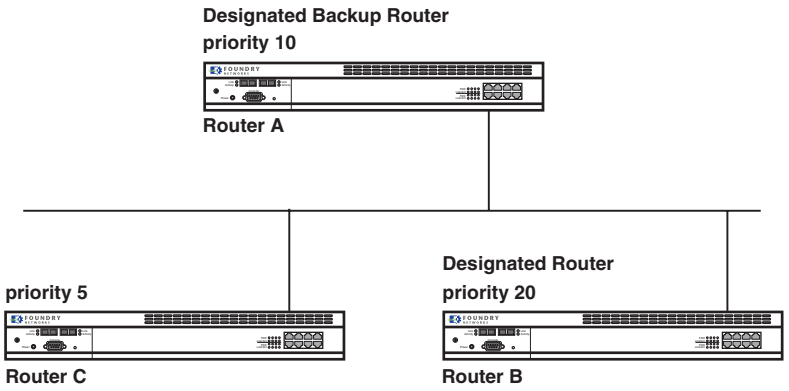
Designated Routers in Multi-Access Networks

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 15.2

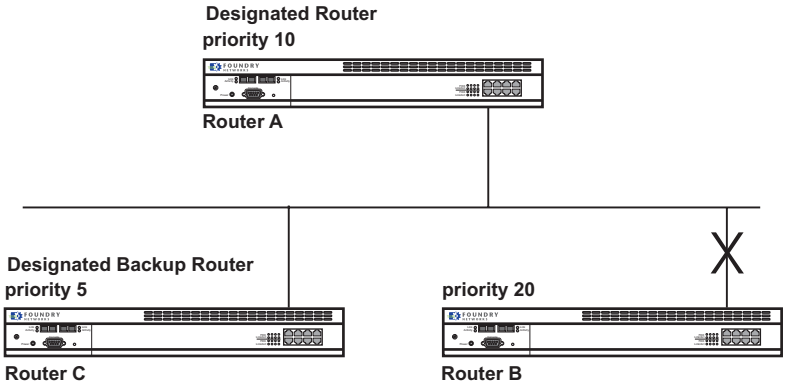
Figure 15.2 Designated and backup router election



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 15.3.

NOTE: Priority is a configurable option at the interface level. You can use this parameter to help bias one router as the DR.

Figure 15.3 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 12-40.

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 15-44.

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).

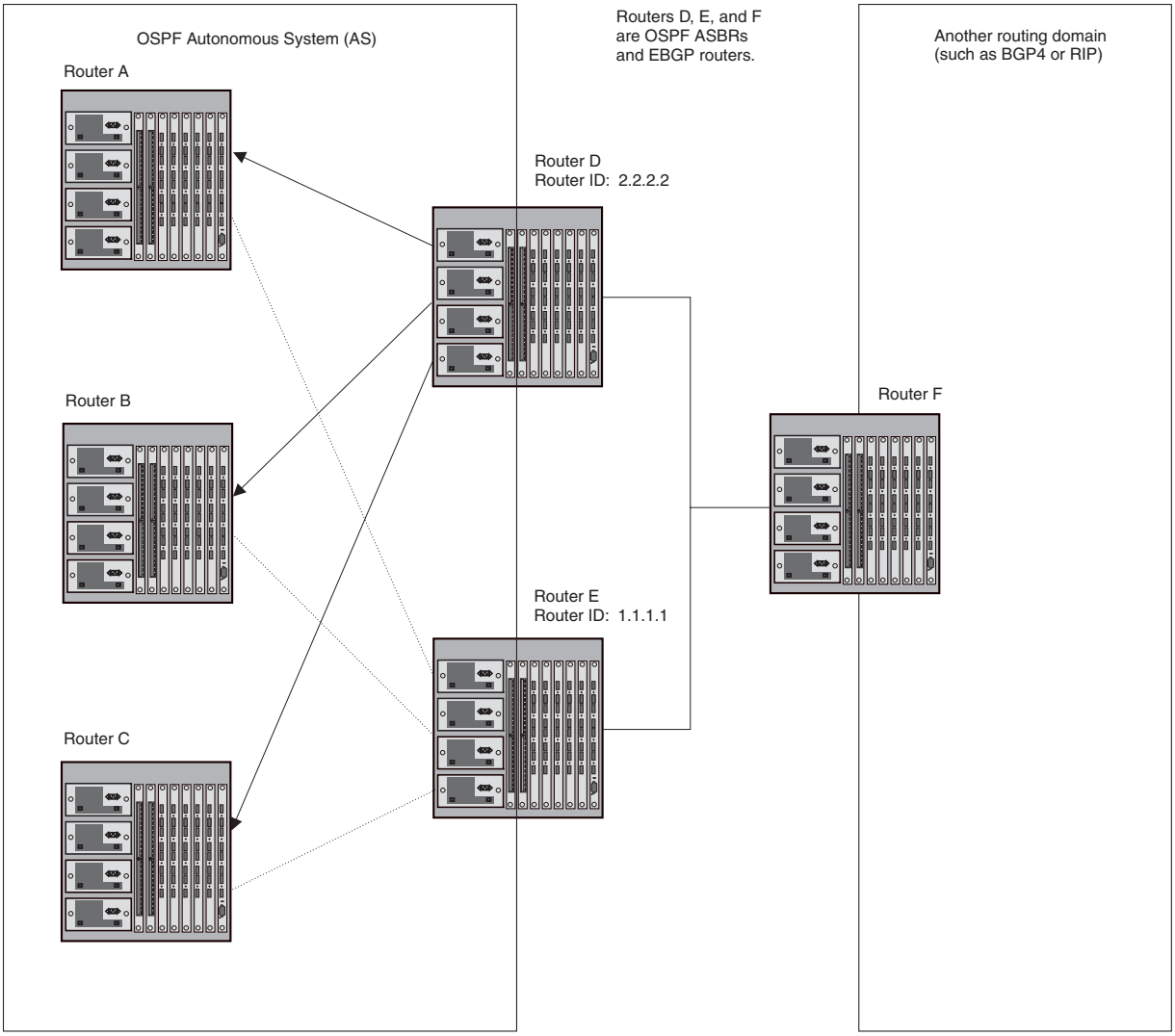
In some cases, multiple ASBRs in an AS can originate equivalent LSAs. The LSAs are equivalent when they have the same cost, the same next hop, and the same destination. Software release 07.1.00 optimizes OSPF by

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 15.4 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 15.4 AS External LSA reduction



Notice that both Router D and Router E have a route to the other routing domain through Router F. In software releases earlier than 07.1.00, 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 15.4, 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 15.4 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

Software release 07.5.00 and later provides support for Appendix E in OSPF RFC 2328. Appendix E describes a method to ensure that an OSPF router (such as a Foundry 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

Software release 07.1.00 and higher dynamically allocate memory for Link State Advertisements (LSAs) and other OSPF data structures.

In previous software releases, OSPF memory is statically allocated. If the Layer 3 Switch runs out of memory for a given LSA type in releases earlier than 07.1.00, an overflow condition occurs and the software sends a message to the Syslog. To change memory allocation requires entering CLI commands and reloading the software.

Software release 07.1.00 and later eliminate the overflow conditions and do 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.

Since dynamic memory allocation is automatic and requires no configuration, the following CLI commands and equivalent Web management options are not supported in software release 07.1.00:

- **maximum-number-of-lsa external** <num>
- **maximum-number-of-lsa router** <num>
- **maximum-number-of-lsa network** <num>
- **maximum-number-of-lsa summary** <num>
- **max-routes** <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 OSPF. The first time you save the device's running configuration (running-config) to the startup-config file, the commands are removed from the file.

NOTE: The **external-lsdb-overflow** command is still supported in accordance with RFC 1765.

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

- 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.
- Enable and configure graceful restart

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, use one of the following methods:

USING THE CLI

```
BigIron(config)# router ospf
```

This command launches you into the OSPF router level where you can assign areas and modify OSPF global parameters.

USING THE WEB MANAGEMENT INTERFACE

1. Log on to the device using a valid user name and password for read-write access. The System configuration panel is displayed.
2. Select Enable next to OSPF.
3. Click the Apply button to save the change to the device's running-config file.

4. 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.

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.

The CLI displays a warning message such as the following:

```
BigIron(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.

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.

EXAMPLE:

To set up the OSPF areas shown in Figure 15.1 on page 15-2, use one of the following methods.

USING THE CLI

```
BigIron(config-ospf-router)# area 192.5.1.0
BigIron(config-ospf-router)# area 200.5.0.0
BigIron(config-ospf-router)# area 195.5.0.0
BigIron(config-ospf-router)# area 0.0.0.0
```



```
BigIron(config-ospf-router) write memory
```

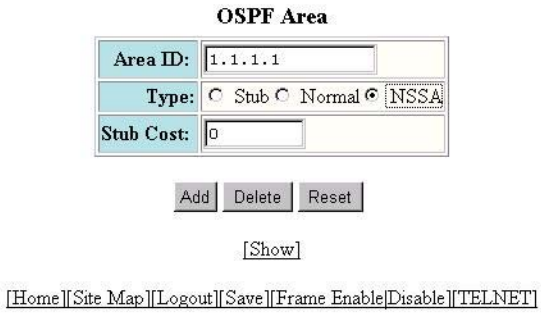
Syntax: area <num> | <ip-addr>

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.

USING THE WEB MANAGEMENT INTERFACE

1. Log on to the device using a valid user name and password for read-write access.
2. If you have not already enabled OSPF, enable it by clicking on the Enable radio button next to OSPF on the System configuration panel, then clicking Apply to apply the change.
3. Click on the plus sign next to Configure in the tree view to expand the list of configuration options.
4. Click on the plus sign next to OSPF in the tree view to expand the list of OSPF option links.
5. Click on the Area link to display the OSPF Area configuration panel, as shown in the following figure.



NOTE: If the device already has OSPF areas, a table listing the areas is displayed. Click the Modify button to the right of the row describing an area to change its configuration, or click the Add Area link to display the OSPF Area configuration panel.

6. Enter the area ID in the Area ID field. The ID can be a number or an IP address.
7. Select the area type by clicking on the radio button next to its description in the Type field. For example, to select NSSA, click next to NSSA.
8. If you are configuring a stub area or NSSA, enter a cost in the Stub Cost field. The parameter is required for those area types but is not required for normal areas. You can specify from 1 – 16777215. There is no default.
9. Click the Add button to add the area to the running-config file.
10. 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.

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 (LSA) 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, use the following CLI method.

USING THE CLI

To disable summary LSAs for a stub area, enter commands such as the following:

```
BigIron(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.

USING THE WEB MANAGEMENT INTERFACE

You can configure a stubby area using the Web management interface, but you cannot disable summary LSAs for the area. You must use the CLI to disable the summary LSAs.

Assign a Not-So-Stubby Area (NSSA)

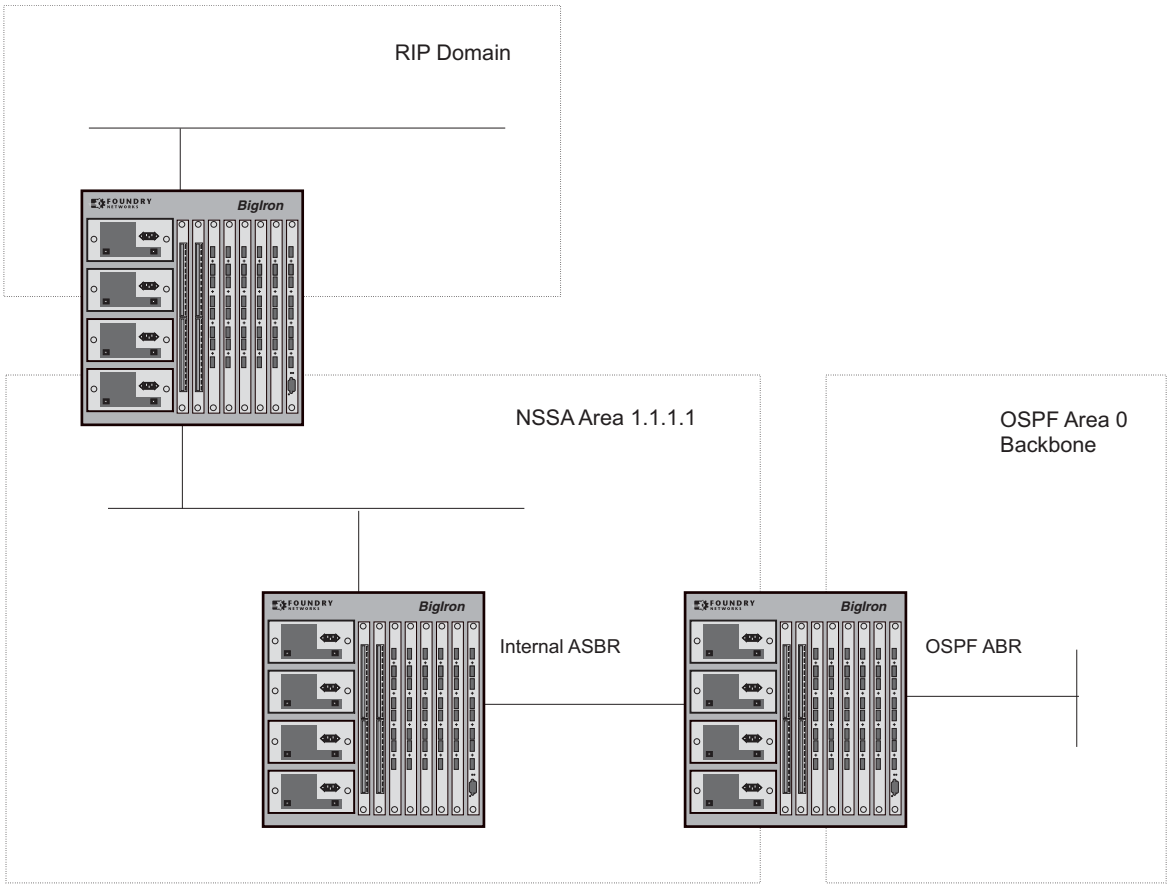
The OSPF Not So Stubby Area (NSSA) feature enables you to configure OSPF areas that provide the benefits of stub areas, but that also are capable of importing external route information. OSPF does not flood external routes from other areas into an NSSA, but does translate and flood route information 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 15.5 shows an example of an OSPF network containing an NSSA.

Figure 15.5 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 an NSSA, use one of the following methods.

USING THE CLI

To configure OSPF area 1.1.1.1 as an NSSA, enter the following commands.

```
BigIron(config)# router ospf
BigIron(config-ospf-router)# area 1.1.1.1 nssa 1
BigIron(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.

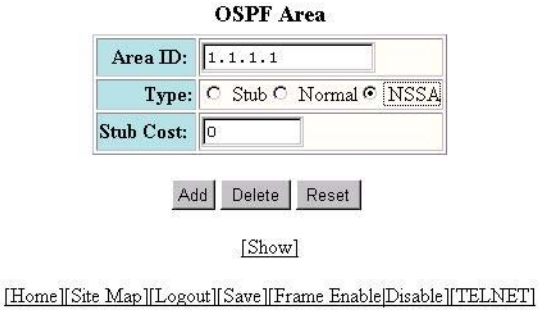
NOTE: The Layer 3 Switch does not inject the default route into an NSSA by default.

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.

USING THE WEB MANAGEMENT INTERFACE

1. Log on to the device using a valid user name and password for read-write access.
2. If you have not already enabled OSPF, enable it by clicking on the Enable radio button next to OSPF on the System configuration panel, then clicking Apply to apply the change.
3. Click on the plus sign next to Configure in the tree view to expand the list of configuration options.
4. Click on the plus sign next to OSPF in the tree view to expand the list of OSPF option links.
5. Click on the Area link to display the OSPF Area configuration panel, as shown in the following figure.



NOTE: If the device already has OSPF areas, a table listing the areas is displayed. Click the Modify button to the right of the row describing an area to change its configuration, or click the Add Area link to display the OSPF Area configuration panel.

6. Enter the area ID in the Area ID field. The ID can be a number or an IP address.
7. Select NSSA by clicking on the radio button next to NSSA in the Type field.
8. Enter a cost in the Stub Cost field. This parameter is required. You can specify from 1 – 16777215. There is no default.
9. Click the Add button to add the area.
10. 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.

Configuring an Address Range 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 an address range. The ABR creates an aggregate value based on the address range. The aggregate value becomes the address that the ABR advertises instead of advertising the individual addresses represented by the aggregate. You can configure up to 32 ranges in an OSPF area.

USING THE CLI

To configure an address range in NSSA 1.1.1.1, enter the following commands. This example assumes that you have already configured NSSA 1.1.1.1.

```
BigIron(config)# router ospf
BigIron(config-ospf-router)# area 1.1.1.1 range 209.157.22.1 255.255.0.0
BigIron(config-ospf-router)# write memory
```

Syntax: [no] area <num> | <ip-addr> range <ip-addr> <ip-mask> [advertise | not-advertise]

The <num> | <ip-addr> parameter specifies the area number, which can be in IP address format. If you specify a number, the number can be from 0 – 2,147,483,647.

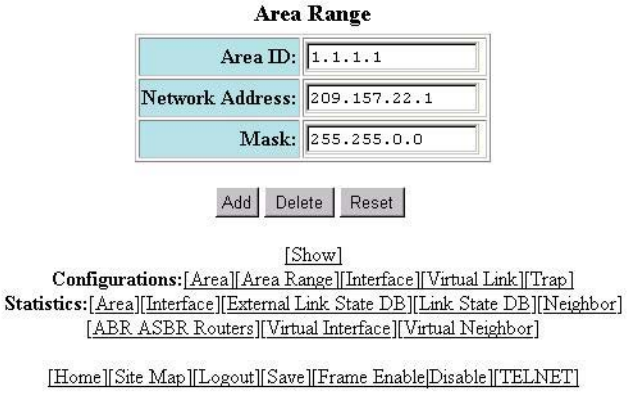
The **range** <ip-addr> parameter specifies the IP address portion of the range. The software compares the address with the significant bits in the mask. All network addresses that match this comparison are summarized in a single route advertised by the router.

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 209.157 are summarized into a single route.

The **advertise | not-advertise** parameter specifies whether you want the Layer 3 Switch to send type 3 LSAs for the specified range in this area. The default is **advertise**.

USING THE WEB MANAGEMENT INTERFACE

1. Log on to the device using a valid user name and password for read-write access.
2. If you have not already enabled OSPF, enable it by clicking on the Enable radio button next to OSPF on the System configuration panel, then clicking Apply to apply the change.
3. Click on the plus sign next to Configure in the tree view to expand the list of configuration options.
4. Click on the plus sign next to OSPF in the tree view to expand the list of OSPF option links.
5. Click on the Area Range link to display the OSPF Area Range configuration panel.
6. Click on the Add Area Range link to display the following panel.



NOTE: If the device already has an OSPF area range, a table listing the ranges is displayed. Click the Modify button to the right of the row describing a range to change its configuration, or click the Add Area Range link to display the OSPF Area Range configuration panel.

7. Enter the area ID in the Area ID field.
8. Enter an IP address in the Network Address field.

9. Enter a network mask in the Mask field. 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.
10. Click the Add button to add the area.
11. 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.

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.

USING THE CLI

EXAMPLE:

To define an area range for subnets on 193.45.5.1 and 193.45.6.2, enter the following command:

```
BigIron(config)# router ospf
BigIron(config-ospf-router)# area 192.45.5.1 range 193.45.0.0 255.255.0.0
BigIron(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.

USING THE WEB MANAGEMENT INTERFACE

1. Log on to the device using a valid user name and password for read-write access.
2. If you have not already enabled OSPF, enable it by clicking on the Enable radio button next to OSPF on the System configuration panel, then clicking Apply to apply the change.
3. Click on the plus sign next to Configure in the tree view to expand the list of configuration options.
4. Click on the plus sign next to OSPF in the tree view to expand the list of OSPF option links.
5. Click on the Area Range link to display the OSPF Area Range configuration panel.
6. Click on the Add Area Range link to display the Area Range panel.

NOTE: If the device already has an OSPF area range, a table listing the ranges is displayed. Click the Modify button to the right of the row describing a range to change its configuration, or click the Add Area Range link to display the OSPF Area Range configuration panel.

---

7. Enter the area ID in the Area ID field.
8. Enter an IP address in the Network Address field.
9. Enter a network mask in the Mask field. 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.
10. Click the Add button to add the area.
11. 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.

Assigning Interfaces to an Area

Once you define OSPF areas, you can assign interfaces 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 8 of Router A to area 192.5.0.0 and then save the changes, use one the following methods:

USING CLI

To assign interface 1/8 of Router A to area 192.5.0.0 and then save the changes, enter the following commands:

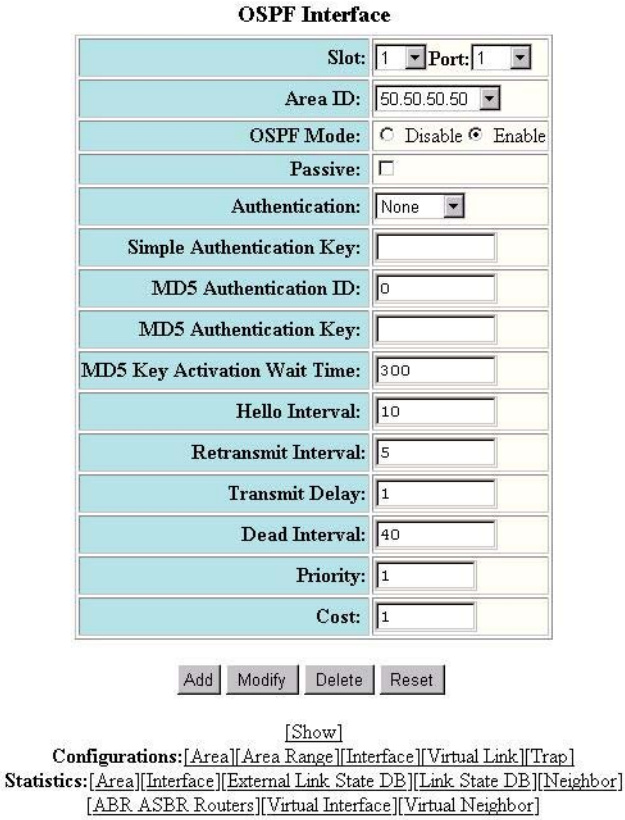
```
RouterA(config-ospf-router)# interface e 1/8
RouterA(config-if-1/8)# ip ospf area 192.5.0.0
RouterA(config-if-1/8)# write memory
```

USING WEB MANAGEMENT INTERFACE

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 an interface to an area:

1. Log on to the device using a valid user name and password for read-write access.
2. If you have not already enabled OSPF, enable it by clicking on the Enable radio button next to OSPF on the System configuration panel, then clicking Apply to apply the change.
3. Click on the plus sign next to Configure in the tree view to expand the list of configuration options.
4. Click on the plus sign next to OSPF in the tree view to expand the list of OSPF option links.
5. Click on the Interface link.
   - If the device does not have any OSPF interfaces, the OSPF Interface configuration panel is displayed, as shown in the following example.
   - If an OSPF interface is already configured and you are adding a new one, click on the Add OSPF Interface link to display the OSPF Interface configuration panel, as shown in the following example.
   - If you are modifying an existing OSPF interface, click on the Modify button to the right of the row describing the interface to display the OSPF Interface configuration panel, as shown in the following example.



1. Select the port (and slot if applicable) to be assigned to the area from the Port and Slot pulldown menus.

NOTE: If you are configuring a Chassis device, a Slot Number pulldown menu will appear on the configuration panel in addition to the Port pulldown menu.

---

2. Select the IP address of the area to which the interface is to be assigned from the Area ID pull down menu.

NOTE: You must configure the area before you can assign interfaces to it.

---

3. Select the Enable option of the OSPF mode parameter to enable OSPF on the interface.
4. Click the Add button to save the change to the device's running-config file.
5. 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.

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.

USING THE CLI

Port default values can be modified using the following CLI 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.

USING THE WEB MANAGEMENT INTERFACE

To modify OSPF port parameters when using the Web:

1. Log on to the device using a valid user name and password for read-write access.
2. If you have not already enabled OSPF, enable it by clicking on the Enable radio button next to OSPF on the System configuration panel, then clicking Apply to apply the change.
3. Click on the plus sign next to Configure in the tree view to expand the list of configuration options.
4. Click on the plus sign next to OSPF in the tree view to expand the list of OSPF option links.
5. Click on the Interface link.

NOTE: If the device already has OSPF interfaces, a table listing the interfaces is displayed. Click the Modify button to the right of the row describing the interface to change its configuration, or click the Add OSPF Interface link to display the OSPF Interface configuration panel.

---

6. Select the port (and slot if applicable) from the pulldown menu(s).
7. Select the area ID from the Area ID pulldown menu.
8. Select the OSPF mode to enable or disable OSPF on the interface.
9. Click on the checkbox next to Passive if you do not want the interface to 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.
10. Select the authentication method for the interface from the pulldown menu. Options are None, Simple, or MD5.

NOTE: If you select MD5 as the authentication method, enter a value for the MD5 authentication ID, key and key activation time in the associated fields. If you select Simple, enter an authentication key. If you select No Authentication as the authentication method, you do not need to specify anything in the Simple and MD5 fields.

---

11. Modify the default values of the following interface parameters as needed: hello interval, retransmit interval, transmit delay, dead interval, priority, and cost.
12. Click the Add button (if you are adding a new neighbor) or the Modify button (if you are modifying a neighbor that is already configured) to apply the changes to the device's running-config file.
13. Select the Save link at the bottom of the dialog, then select Yes when prompted to save the configuration change to the startup-config file on the device's flash memory.

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 12-20.

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, software release 07.1.10 and later encrypts 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

USING THE CLI

To change the authentication-change interval, enter a command such as the following at the interface configuration level of the CLI:

```
BigIron(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.

USING THE CLI

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.

```
BigIron(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:

```
BigIron(config-if-1/1)# no ip ospf database-filter all out
```

USING THE WEB MANAGEMENT INTERFACE

You cannot configure filters to block flooding on OSPF interfaces using the Web management interface.

Configuring an OSPF Non-Broadcast Interface

In software releases 08.0.00 and FES 03.4.00, you can 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 configure non-broadcast and neighbor. 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. This feature is not supported on POS or ATM interfaces.

To configure an OSPF interface as a non-broadcast interface, you 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:

```
BigIron(config)# int ve 20
BigIron(config-vif-20)# ip ospf area 0
BigIron(config-vif-20)# ip ospf network non-broadcast
BigIron(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 sub-net as a non-broadcast interface.

```
BigIron(config)# router ospf
BigIron(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 sub-net.

For example:

```
BigIron# 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 sub-net. If no neighbors are configured in the same sub-net, 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.

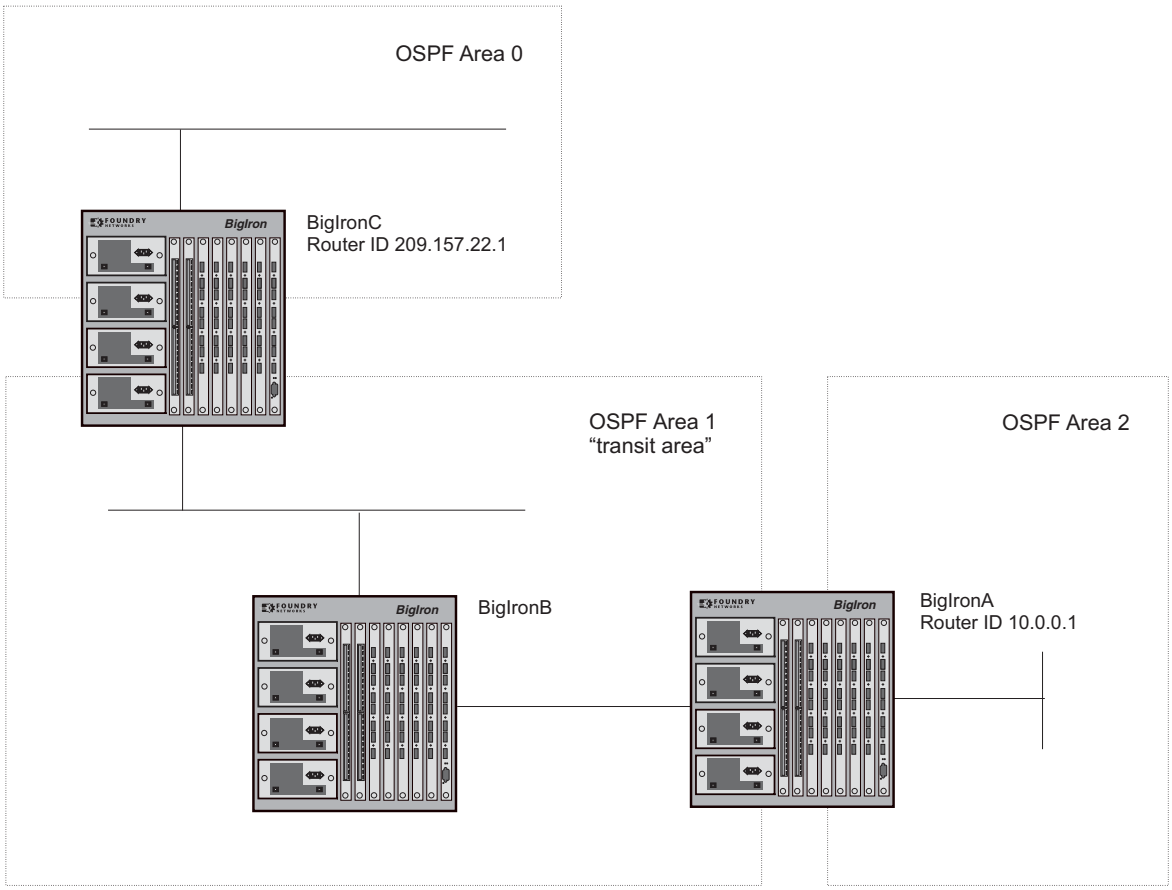
Two parameters fields must be defined for all virtual links—transit area ID and neighbor router.

- 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 12-40.

NOTE: When you establish an area virtual link, you must configure it on both of the routers (both ends of the virtual link).

Figure 15.6 Defining OSPF virtual links within a network



USING THE CLI

EXAMPLE:

Figure 15.6 shows an OSPF area border router, BigIronA, that is cut off from the backbone area (area 0). To provide backbone access to BigIronA, you can add a virtual link between BigIronA and BigIronC 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 BigIronA, enter the following commands:

```
BigIronA(config-ospf-router)# area 1 virtual-link 209.157.22.1
BigIronA(config-ospf-router)# write memory
```

Enter the following commands to configure the virtual link on BigIronC:

```
BigIronC(config-ospf-router)# area 1 virtual-link 10.0.0.1
BigIronC(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.

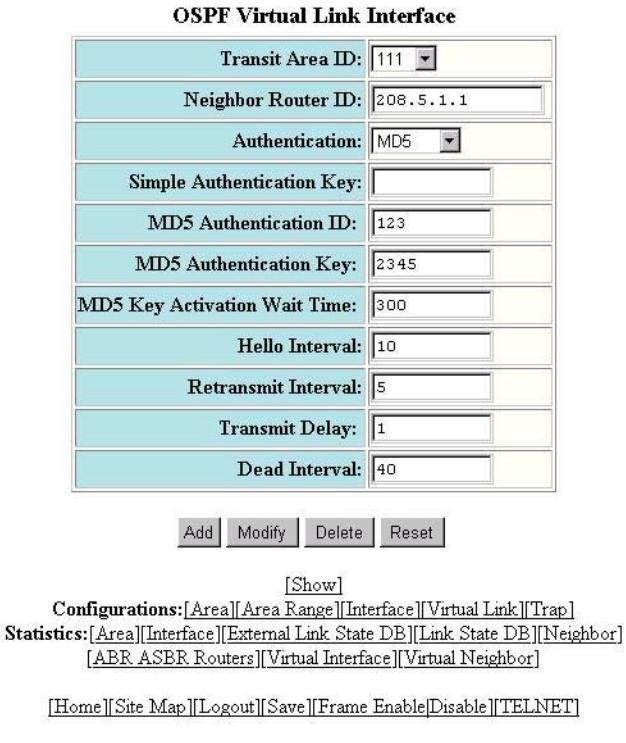
The <router-id> parameter specifies the router ID of the OSPF router at the remote end of the virtual link. To display the router ID on a Foundry Layer 3 Switch, enter the **show ip** command.

See "Modify Virtual Link Parameters" on page 15-26 for descriptions of the optional parameters.

USING THE WEB MANAGEMENT INTERFACE

To configure a virtual link:

1. Log on to the device using a valid user name and password for read-write access.
2. If you have not already enabled OSPF, enable it by clicking on the Enable radio button next to OSPF on the System configuration panel, then clicking Apply to apply the change.
3. Click on the plus sign next to Configure in the tree view to expand the list of configuration options.
4. Click on the plus sign next to OSPF in the tree view to expand the list of OSPF option links.
5. Click on the Virtual Link link.
   - If the device does not have any OSPF virtual links, the OSPF Virtual Link Interface configuration panel is displayed, as shown in the following example.
   - If an OSPF virtual link is already configured and you are adding a new one, click on the Add OSPF Virtual Link link to display the OSPF Virtual Link Interface configuration panel, as shown in the following example.
   - If you are modifying an existing OSPF virtual link, click on the Modify button to the right of the row describing the virtual link to display the OSPF Virtual Link Interface configuration panel, as shown in the following example.



6. Select the transit area ID from the pulldown menu. The transit area is the area ID of the area shared by both routers.
7. Select an authentication method from the pulldown menu. If you select Simple, enter the authentication key in the appropriate field. If you select MD5, enter the MD5 authentication ID, key, and wait time.

NOTE: For descriptions of the authentication parameters, see “Modify Virtual Link Parameters” on page 15-26.

8. Enter the router ID of the neighbor router.
9. Modify the default settings of the following parameters if needed: hello interval, transit delay, retransmit interval and, dead interval.

NOTE: For a description of all virtual link parameters and their possible values, see “Modify Virtual Link Parameters” on page 15-26.

10. Click Add to save the change to the device’s running-config file.
11. Select the Save link at the bottom of the dialog, then select Yes when prompted to save the configuration change to the startup-config file on the device’s flash memory.
12. Log onto the neighbor router and configure the other end of the virtual link.

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.

USING THE CLI

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*.

USING THE WEB MANAGEMENT INTERFACE

To modify virtual link default values:

1. Log on to the device using a valid user name and password for read-write access.
2. Click on the plus sign next to Configure in the tree view to expand the list of configuration options.
3. Click on the plus sign next to OSPF in the tree view to expand the list of OSPF option links.
4. Click on the Virtual Link link to display a table listing the virtual links.
5. Click on the Modify button to the right of the row describing the virtual link you want to modify. The OSPF Virtual Link Interface configuration panel is displayed.
6. Modify the parameters as needed. (See the following section for descriptions of the parameters.)
7. Click Add to save the change to the device’s running-config file.
8. Select the Save link at the bottom of the dialog, then select Yes when prompted to save the configuration change to the startup-config file on the device’s flash memory.
9. Log on to the neighbor router and configure parameter changes to match those configured for the local router.

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, software release 07.1.10 and later encrypts 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

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 reference bandwidth, use the following CLI method.

USING THE CLI

To change the reference bandwidth, enter a command such as the following at the OSPF configuration level of the CLI:

```
BigIron(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, which results in the same costs as previous software releases.

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:

```
BigIron(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 15.7 on page 15-30, an administrator wants to configure the BigIron 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.

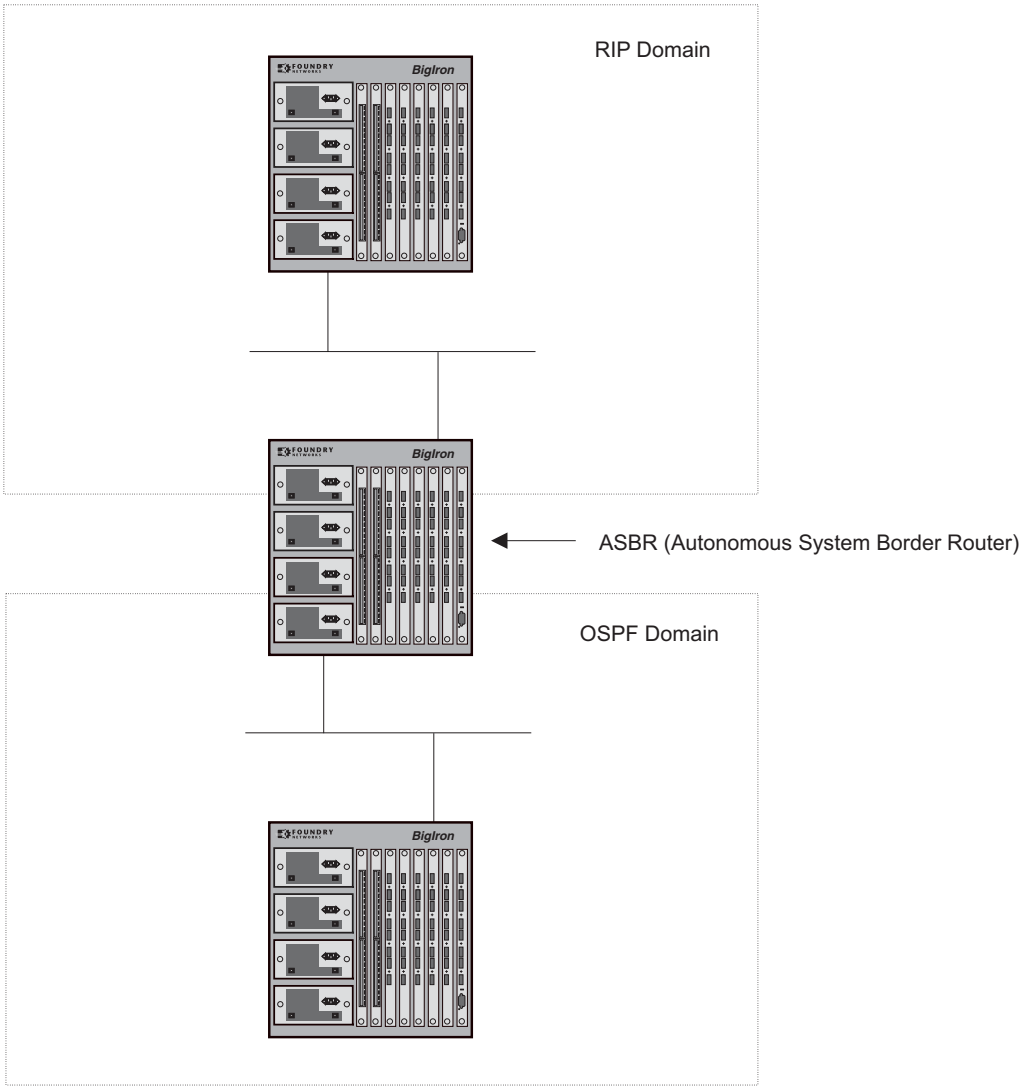
NOTE: The ASBR must be running both RIP and OSPF protocols to support this activity.

To configure for redistribution, define the redistribution tables with deny and permit redistribution filters.

- If you are using the CLI, use the **deny** and **permit** redistribute commands for OSPF at the OSPF router level.
- If you are using the Web management interface, click on the plus sign next to Configure in the tree view, click on the plus sign next to OSPF, then select the Redistribution Filter link from the OSPF configuration sheet.

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 15.7 Redistributing OSPF and static routes to RIP routes



USING THE CLI

EXAMPLE:

To configure the BigIron Layer 3 Switch acting as an ASBR in Figure 15.7 to redistribute OSPF, BGP4, and static routes into RIP, enter the following commands:

```
BigIronASBR(config)# router rip
BigIronASBR(config-rip-router)# permit redistribute 1 all
BigIronASBR(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>]]]

EXAMPLE:

To redistribute RIP, static, and BGP4 routes into OSPF, enter the following commands on the Layer 3 Switch acting as an ASBR:

```
BigIronASBR(config)# router ospf
BigIronASBR(config-ospf-router)# permit redistribute 1 all
BigIronASBR(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.

```
BigIron(config)# router ospf
BigIron(config-ospf-router)# redistribution rip
BigIron(config-ospf-router)# redistribution static
BigIron(config-ospf-router)# write memory
```

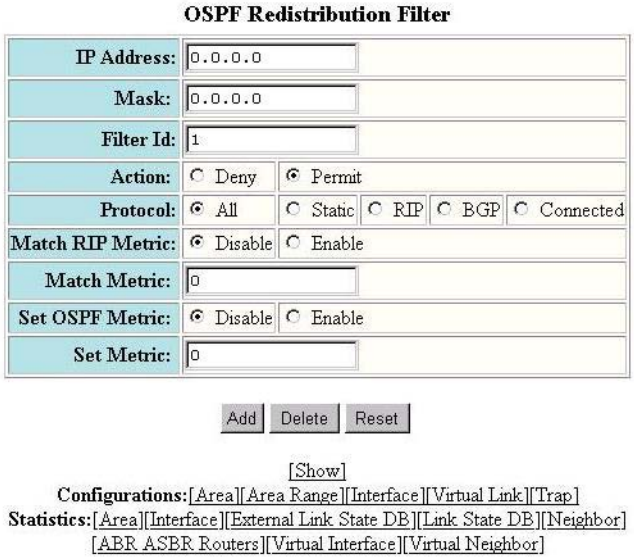
NOTE: The **redistribution** command does not perform the same function as the **permit redistribute** and **deny redistribute** commands. The **redistribute** commands allow you to control redistribution of routes by filtering on the IP address and network mask of a route. The **redistribution** commands 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.

USING THE WEB MANAGEMENT INTERFACE

1. Log on to the device using a valid user name and password for read-write access. The System configuration panel is displayed.
2. Click on the plus sign next to Configure in the tree view to expand the list of configuration options.
3. Click on the plus sign next to OSPF in the tree view to expand the list of OSPF option links.
4. Click on the Redistribution Filter link.
   - If the device does not have any OSPF redistribution filters, the OSPF Redistribution Filter configuration panel is displayed, as shown in the following example.
   - If an OSPF redistribution filter is already configured and you are adding a new one, click on the Add Redistribution Filter link to display the OSPF Redistribution Filter configuration panel, as shown in the following example.

5. If you are modifying an existing OSPF redistribution filter, click on the Modify button to the right of the row describing the filter to display the OSPF Redistribution Filter configuration panel, as shown in the following example.



6. Optionally, enter the IP address and mask if you want to filter the redistributed routes for a specific network range.
7. Optionally, enter the filter ID or accept the ID value in the Filter ID field.
8. Optionally, select the filter action, Deny or Permit. The default is Permit.
9. Optionally, select the types of routes the filter applies to in the Protocol section. You can select one of the following:
   - All (the default)
   - Static
   - RIP
   - BGP
   - Connected
10. Optionally, enable matching on RIP metric and enter the metric.
11. Optionally, enable setting the OSPF metric for the imported routes and specify the metric.
12. Click the Add button to apply the filter to the device's running-config file.
13. Select the Save link at the bottom of the dialog, then select Yes when prompted to save the configuration change to the startup-config file on the device's flash memory.

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.

USING 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:

```
BigIron(config)# ip access-list standard no_ip
BigIron(config-std-nacl)# deny 4.0.0.0 0.255.255.255
BigIron(config-std-nacl)# permit any any
BigIron(config-std-nacl)# exit
BigIron(config)# router ospf
BigIron(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 type> parameter is available in releases 07.6.04, 09.1.00, 03.1.00, and later releases of these branch releases. It identifies the interface type (i.e., **e** (ethernet) or **ve** (virtual)) on which to apply the ACL.

The <interface number> parameter is available in releases 07.6.04, 09.1.00, 03.1.00, and later releases of these branch releases. It 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

To use an extended ACL to configure an OSPF distribution list for denying specific routes, enter commands such as the following:

```
BigIron(config)# ip access-list extended no_ip
BigIron(config-ext-nacl)# deny ip 4.0.0.0 0.255.255.255 255.255.0.0 0.0.255.255
BigIron(config-ext-nacl)# permit ip any any
BigIron(config-ext-nacl)# exit
BigIron(config)# router ospf
BigIron(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**.

The <source-ip> <wildcard> 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 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.

The <destination-ip> <wildcard> parameter specifies the destination address for the policy. Since this ACL is input to an OSPF distribution list, the <destination-ip> parameter actually is specifying the network mask of the destination. The <wildcard> parameter specifies the portion of the destination address to match against. 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.

USING THE CLI

To assign a default metric of 4 to all routes imported into OSPF, enter the following commands:

```
BigIron(config)# router ospf
BigIron(config-ospf-router)# default-metric 4
```

Syntax: default-metric <value>

The <value> can be from 1 – 16,777,215. The default is 10.

USING THE WEB MANAGEMENT INTERFACE

To modify the cost that is assigned to redistributed routes:

1. Log on to the device using a valid user name and password for read-write access.
2. Click on the plus sign next to Configure in the tree view to expand the list of configuration options.
3. Click on the plus sign next to OSPF in the tree view to expand the list of OSPF option links.
4. Click on the Redistribution Filter link to display a table listing the redistribution filters.
5. Click on the Modify button to the right of the row describing the virtual link you want to modify. The OSPF Virtual Link Interface configuration panel is displayed.
6. Enter a value from 1 – 15 in the Default Metric field.
7. Click Add to save the change to the device's running-config file.
8. Select the Save link at the bottom of the dialog, then select Yes when prompted to save the configuration change to the startup-config file on the device's flash memory.

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.

USING THE CLI

To enable redistribution of RIP and static IP routes into OSPF, enter the following commands.

```
BigIron(config)# router ospf
BigIron(config-ospf-router)# redistribution rip
BigIron(config-ospf-router)# redistribution static
BigIron(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:

```
BigIron(config)# ip route 1.1.0.0 255.255.0.0 207.95.7.30
BigIron(config)# ip route 1.2.0.0 255.255.0.0 207.95.7.30
BigIron(config)# ip route 1.3.0.0 255.255.0.0 207.95.7.30
BigIron(config)# ip route 4.1.0.0 255.255.0.0 207.95.6.30
BigIron(config)# ip route 4.2.0.0 255.255.0.0 207.95.6.30
BigIron(config)# ip route 4.3.0.0 255.255.0.0 207.95.6.30
BigIron(config)# ip route 4.4.0.0 255.255.0.0 207.95.6.30 5
BigIron(config)# route-map abc permit 1
BigIron(config-routemap abc)# match metric 5
BigIron(config-routemap abc)# set metric 8
BigIron(config-routemap abc)# router ospf
BigIron(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.

```
BigIron(config-ospf-router)# show ip ospf database 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>

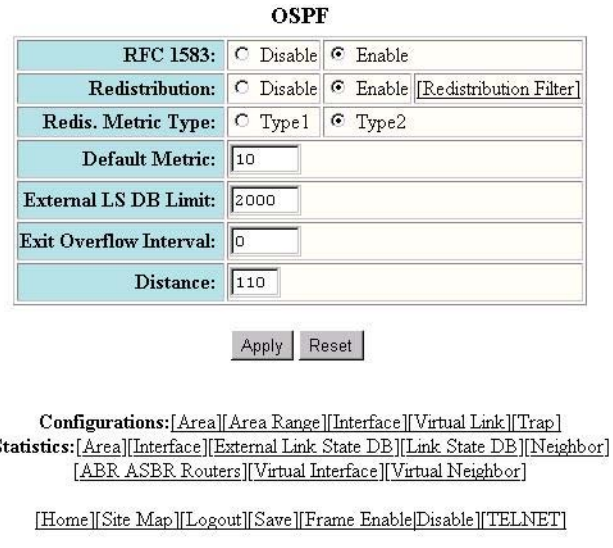
NOTE: You must configure the route map before you configure a redistribution filter that uses the route map.

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.

USING THE WEB MANAGEMENT INTERFACE

1. Log on to the device using a valid user name and password for read-write access. The System configuration panel is displayed.
2. Click on the plus sign next to Configure in the tree view to expand the list of configuration options.
3. Click on the plus sign next to OSPF in the tree view to expand the list of BGP option links.
4. Click on the General link to display the OSPF configuration panel, as shown in the following figure.



5. Select the Enable radio button next to Redistribution.
6. Click the Apply button to apply the change to the device's running-config file.

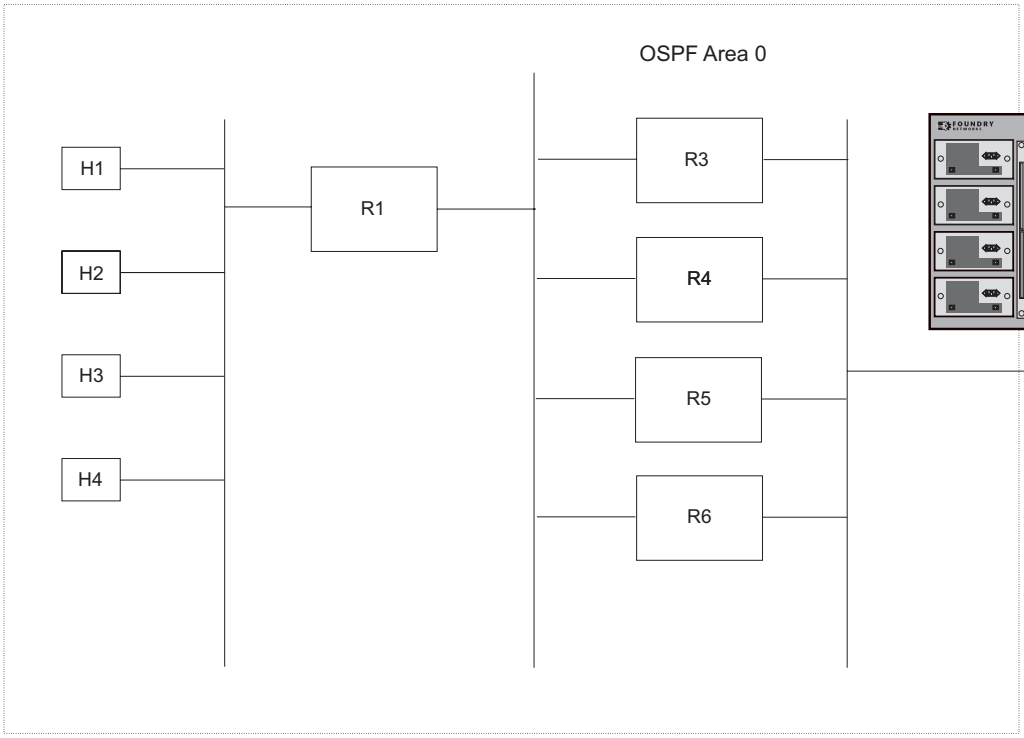
7. Select the Save link at the bottom of the dialog, then select Yes when prompted to save the configuration change to the startup-config file on the device's flash memory.

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 – 8 paths.

The router software can use the route information it learns through OSPF to determine the paths and costs. Figure 15.8 shows an example of an OSPF network containing multiple paths to a destination (in this case, R1).

Figure 15.8 Example OSPF network with four equal-cost paths



In the example in Figure 15.8, the Foundry router has four paths to R1:

- BI->R3
- BI->R4
- BI->R5
- BI->R6

Normally, the Foundry router 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 router 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 router now has four equal-cost paths to R1. To allow the router 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 – 8 paths.

NOTE: The Foundry router is not source routing in these examples. The router 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 12-66.

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.

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. Type 7-route redistribution is not affected by this feature. All type 7 routes will be imported (if redistribution is enabled). To summarize type 7 LSAs or exported routes, use NSSA address range summarization.

To configure route summarization, use the following CLI method.

USING THE CLI

To configure a summary address for OSPF routes, enter commands such as the following:

```
BigIron(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:

```
BigIron(config-ospf-router)# 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

USING THE WEB MANAGEMENT INTERFACE

You cannot configure OSPF route summarization using the Web management interface.

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 15-12.

To enable default route origination, use the following CLI method.

USING THE CLI

To enable default route origination, enter the following command:

```
BigIron(config-ospf-router)# default-information-originate
```

To disable the feature, enter the following command:

```
BigIron(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.

USING THE WEB MANAGEMENT INTERFACE

You cannot configure OSPF default route origination using the Web management interface.

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, use the following CLI method.

USING THE CLI

To change the SPF delay and hold time, enter commands such as the following:

```
BigIron(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:

```
BigIron(config-ospf-router)# no timers spf 10 20
```

USING THE WEB MANAGEMENT INTERFACE

You cannot configure the SPF timers using the Web management interface.

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.

USING THE CLI

To modify the default value to type 1, enter the following command:

```
BigIron(config-ospf-router)# metric-type type1
```

Syntax: metric-type type1 | type2

The default is **type2**.

USING THE WEB MANAGEMENT INTERFACE

To modify the default metric type:

1. Log on to the device using a valid user name and password for read-write access. The System configuration panel is displayed.
2. Click on the plus sign next to Configure in the tree view to expand the list of configuration options.
3. Click on the plus sign next to OSPF in the tree view to expand the list of OSPF option links.
4. Click on the General link to display the OSPF configuration panel.
5. Select either Type 1 or Type 2 for the redistribution metric type.
6. Click the Apply button to save the change to the device's running-config file.
7. 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.

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 16-38 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 configure administrative distances for OSPF route types, use the following CLI method.

USING THE CLI

To change the default administrative distances for inter-area routes, intra-area routes, and external routes, enter the following command:

```
BigIron(config-ospf-router)# distance external 100
BigIron(config-ospf-router)# distance inter-area 90
BigIron(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:

```
BigIron(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, use the following CLI method.

USING THE CLI

To change the LSA pacing interval to two minutes (120 seconds), enter the following command:

```
BigIron(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:

```
BigIron(config-ospf-router)# no timers lsa-group-pacing
```

USING THE WEB MANAGEMENT INTERFACE

You cannot configure this option using the Web management interface.

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.

USING THE CLI

When using the CLI, you can disable all or specific OSPF trap generation by entering the following CLI command:

```
BigIron(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

EXAMPLE:

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:

```
BigIron(config-ospf-router)# no trap neighbor-state-change-trap
```

EXAMPLE:

To reinstate the trap, enter the following command:

```
BigIron(config-ospf-router)# trap neighbor-state-change-trap
```

Syntax: [no] snmp-server trap ospf <ospf-trap>

USING THE WEB MANAGEMENT INTERFACE

To disable a specific OSPF trap or traps:

1. Log on to the device using a valid user name and password for read-write access. The System configuration panel is displayed.
2. Click on the plus sign next to Configure in the tree view to expand the list of configuration options.
3. Click on the plus sign next to OSPF in the tree view to expand the list of OSPF option links.
4. Click on the Trap link to display the OSPF Trap panel.
5. Select the Disable radio button beside each OSPF trap you want to disable.
6. Click the Apply button to save the change to the device's running-config file.
7. 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.

Modify OSPF Standard Compliance Setting

Foundry routers are configured, by default, to be compliant with the RFC 1583 OSPF V2 specification.

USING THE CLI

To configure a router to operate with the latest OSPF standard, RFC 2178, enter the following commands:

```
BigIron(config)# router ospf
BigIron(config-ospf-router)# no rfc1583-compatibility
```

Syntax: [no] rfc1583-compatibility

USING THE WEB MANAGEMENT INTERFACE

To configure a router to operate with the latest OSPF standard, RFC 2178:

1. Log on to the device using a valid user name and password for read-write access. The System configuration panel is displayed.
2. Click on the plus sign next to Configure in the tree view to expand the list of configuration options.
3. Click on the plus sign next to OSPF in the tree view to expand the list of OSPF option links.
4. Click on the General link to display the OSPF configuration panel.
5. Select Disable next to RFC 1583.
6. Click the Apply button to save the change to the device's running-config file.
7. 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.

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: Software release 07.1.00 and later dynamically allocate OSPF memory as needed. See "Dynamic OSPF Memory" on page 15-7.

USING THE CLI

To modify the exit overflow interval to 60 seconds, enter the following command:

```
BigIron(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.

USING THE WEB MANAGEMENT INTERFACE

To modify the exit overflow interval:

1. Log on to the device using a valid user name and password for read-write access. The System configuration panel is displayed.
2. Click on the plus sign next to Configure in the tree view to expand the list of configuration options.
3. Click on the plus sign next to OSPF in the tree view to expand the list of OSPF option links.
4. Click on the General link to display the OSPF configuration panel.
5. Enter a value from 0 – 86400 in the Exit Overflow Interval field.
6. Click the Apply button to save the change to the device's running-config file.
7. 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.

Modify the Maximum Number of Routes

NOTE: This section applies only to devices that are running software earlier than release 07.1.00. See "Dynamic OSPF Memory" on page 15-7.

The OSPF route table holds 16000 routes by default. You can change the maximum number of routes the Layer 3 Switch's OSPF table can hold to a value from 4000 – 32000.

USING THE CLI

To change the maximum number of OSPF routes to 32000, enter the following command:

```
BigIron(config-ospf-router)# max-routes 32000
BigIron(config-ospf-router)# end
BigIron# reload
```

Syntax: `max-routes <num>`

The <num> indicates the number of OSPF routes allowed and can be from 4000 – 32000. The change takes effect after the router is rebooted.

USING THE WEB MANAGEMENT INTERFACE

You cannot modify the maximum number of OSPF routes using the Web management interface.

Modify LSDB Limits

NOTE: This section applies only to devices that are running software earlier than release 07.1.00. See "Dynamic OSPF Memory" on page 15-7.

On Layer 3 Switches with 32MB or greater memory, you can modify the number of link-state advertisements (LSAs) that the router allows before a database overflow condition is declared on the system. These parameters are part of the router's compliance with RFC 1765.

The following table lists the types of LSAs for which you can configure the table sizes, the default number of entries the tables can hold, and the range of maximum values you can specify. You cannot configure the LSA tables globally; you must configure them for individual LSA types. Make sure you save the running-config file and reload after changing a table size. The change does not take effect until you reload or reboot.

Table 15.1: Configurable LSA Table Sizes

LSA Type	Default Maximum Number of Entries	Range of Values
External (type 5)	2000	500 – 8000
Network (type 2)	2000	200 – 2000
Router (type 1)	2200	200 – 2200
Summary (type 3 and type 4)	2000	500 – 8000 (NetIron Stackable) 500 – 18000 (Chassis devices, TurboIron/8)

USING THE CLI

To change the maximum number of summary LSA entries from 2000 to 18000, enter the following commands:

```
BigIron(config-ospf-router)# maximum-number-of-lsa summary 18000
BigIron(config-ospf-router)# write memory
```

Syntax: maximum-number-of-lsa external | network | router | summary <value>

USING THE WEB MANAGEMENT INTERFACE

To modify the number of IP OSPF external link state advertisements:

1. Log on to the device using a valid user name and password for read-write access. The System configuration panel is displayed.

2. Click on the plus sign next to Configure in the tree view to expand the list of configuration options.
3. Click on the plus sign next to OSPF in the tree view to expand the list of OSPF option links.
4. Click on the General link to display the OSPF configuration panel.
5. Enter a value from 500 – 8000 in the External LSDB Limit field.
6. Click the Apply button to save the change to the device's running-config file.
7. 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.

Configuring an OSPF Point-to-Point Link

OSPF point-to-point links are supported in software releases 07.8.00 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 in software releases 07.8.00 and later and on the BigIron MG8 and NetIron 40G, this command is available in software release 02.2.01 and later.
- On devices running Enterprise software release 07.8.00 and later, this feature is supported on 10/100, Gigabit Ethernet, and 10-Gigabit Ethernet interfaces. on the BigIron MG8 and NetIron 40G, this command is available in software release 02.2.01 and later, 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.
- (Enterprise software release 07.8.00 and later only) PoS and ATM links are point-to-point links by default. Therefore, you do not need to enable this feature on these interface types.
- 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:

```
BigIron(config)# interface eth 1/5
BigIron(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 15-54 and "Displaying OSPF Interface Information" on page 15-56.

Specifying Types of OSPF Syslog Messages to Log

NOTE: This section applies only to devices that are running software release 07.6.03 or later.

Starting with release 07.6.03, you can specify which kinds of OSPF-related Syslog messages are logged. In releases prior to 07.6.03, by default all OSPF Syslog messages are logged. In configurations with a large amount of OSPF activity, this can result in the Foundry device's Syslog buffer and the Syslog server filling up with OSPF messages.

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.

For example, to specify that all OSPF-related Syslog messages be logged, enter the following commands.

```
BigIron(config)# router ospf
BigIron(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.

Configuring Graceful Restart

The Graceful Restart feature provides support for high-availability routing.

NOTE: On the BigIron MG8 and NetIron 40G, this feature is available in software release 02.2.01

With this feature enabled, disruptions in forwarding are minimized and route flapping diminished to provide continuous service during times when a router experiences a restart. A restarting router sends special LSAs to its neighbors called **grace-lsas**. These LSAs are sent to neighbors either before a planned OSPF restart or immediately after an unplanned restart. A grace LSA contains a grace period value that the requesting routers asks its neighbor routers to use for the existing routes, to and through the router after a restart. The restarting router comes up, it continues to use its existing OSPF routes to forward packets. In the background, it re-establishes OSPF adjacencies with its neighboring router, relearns all OPSF LSAs, recalculates its OSPF routes, and replaces them with new routes as necessary. Once the restarting router relearns all OSPF routes, it flushes the grace LSAs from the network, informing the helper routers of the completion of the restart process. If the restarting router does not re-establish adjacencies with the helper router within the restart time, the helper router stops the helping function and flushes the stale OSPF routes.

Configuring OSPF Graceful Restart

To configure OSPF Graceful Restart on a router, the restarting router and its directly connected OSPF peers must be enabled with Graceful Restart.

```
BigIron MG8(config)#router ospf
BigIron MG8(config-ospf-router)#area 0
BigIron MG8(config-ospf-router)#graceful-restart
```

Syntax: graceful-restart

Enabling and Disabling OSPF Helper

When OSPF is enabled, the helper mode is enabled by default. OSPF routers that do not have graceful restart enabled will act as if the graceful restart helper is enabled. To prevent the graceful restart from performing its function, disable it by entering the following command:

```
BigIron MG8(config-ospf-router)#graceful-restart helper-disable
```

Syntax: [no] graceful-restart helper disable

Use the **no** form of the command to re-enable the graceful restart helper.

Configuring OSPF Graceful Restart Timer

The OSPF graceful restart timer specifies the maximum amount of time an OSPF restarting router will take to re-establish OSPF adjacencies and relearn OSPF routes. This value will be sent to the neighboring routers in the grace LSA packets. Configure the timer by entering a command such as the following:

```
BigIron MG8(config-ospf-router)#graceful-restart restart-time 120
```

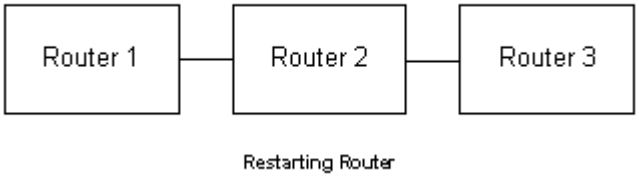
Syntax: graceful-restart restart-time <seconds>

Enter 10 – 1200 for seconds. The default is 120 seconds.

Configuration Examples of Graceful Restart in OSPF

OSPF Graceful Restart requires at least three routers as shown in Figure 15.9.

Figure 15.9 Restarting Router Topology



Before configuring graceful restart, use the **show ip ospf neighbor** command to determine the state of the OSPF neighbors. For example,

```
BigIron MG8 1#show ip ospf neighbor
Port   Address          Pri State      Neigh Address    Neigh ID         Ev Opt Cnt
3/7    40.0.1.1         1   FULL/DR    40.0.1.3         9.0.1.24         23 2   0
```

Enable graceful restart on each OSPF router in Figure 15.9. For example,

Router 1

```
BigIron MG8(config)#router ospf
BigIron MG8(config-ospf-router)#graceful-restart
BigIron MG8(config-ospf-router)#area 0
```

Router 2

```
BigIron MG8(config)#router ospf
BigIron MG8(config-ospf-router)#graceful-restart
BigIron MG8(config-ospf-router)#area 0
```

Router 3

```
BigIron MG8(config)#router ospf
BigIron MG8(config-ospf-router)#graceful-restart
BigIron MG8(config-ospf-router)#area 0
```

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 15-51.
- CPU utilization statistics – see “Displaying CPU Utilization Statistics” on page 15-52.
- Area information – see “Displaying OSPF Area Information” on page 15-53.
- Neighbor information – see “Displaying OSPF Neighbor Information” on page 15-54.
- Interface information – see “Displaying OSPF Interface Information” on page 15-56.
- Route information – see “Displaying OSPF Route Information” on page 15-58.
- External link state information – see “Displaying OSPF External Link State Information” on page 15-60.
- Link state information – see “Displaying OSPF Link State Information” on page 15-61.
- Virtual Neighbor information – see “Displaying OSPF Virtual Neighbor Information” on page 15-63.
- Virtual Link information – see “Displaying OSPF Virtual Link Information” on page 15-63.
- ABR and ASBR information – see “Displaying OSPF ABR and ASBR Information” on page 15-63.
- Trap state information – see “Displaying OSPF Trap Status” on page 15-64.
- “Displaying OSPF Graceful Restart Information” on page 15-64.

Displaying General OSPF Configuration Information

To display general OSPF configuration information, enter the following command at any CLI level:

```
BigIron> 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

USING THE WEB MANAGEMENT INTERFACE

1. Log on to the device using a valid user name and password for read-only or read-write access. The System configuration panel is displayed.
2. Click on the plus sign next to Configure in the tree view to expand the list of configuration options.

3. Click on the plus sign next to OSPF in the tree view to expand the list of OSPF option links.
4. Click on the General link to display the OSPF configuration panel.

Displaying CPU Utilization Statistics

You can display CPU utilization statistics for OSPF and other IP protocols.

USING THE CLI

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:

```
BigIron# 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:

```
BigIron# 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:

```
BigIron# 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.

USING THE WEB MANAGEMENT INTERFACE

You cannot display this information using the Web management interface.

Displaying OSPF Area Information

To display global OSPF area information for the router, use one of the following methods.

USING THE CLI

To display OSPF area information, enter the following command at any CLI level:

```
BigIron> 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 15.2: 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.

Table 15.2: CLI Display of OSPF Area Information (Continued)

This Field...	Displays...
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.

USING THE WEB MANAGEMENT INTERFACE

1. Log on to the device using a valid user name and password for read-only or read-write access. The System configuration panel is displayed.
2. Click on the plus sign next to Monitor in the tree view to expand the list of monitoring options.
3. Click on the plus sign next to OSPF in the tree view to expand the list of OSPF option links.
4. Click on the Area link.

Displaying OSPF Neighbor Information

To display OSPF neighbor information for the router, use one of the following methods.

NOTE: Releases prior to 07.6.03 support the command **show ip ospf neighbor** only. However, this command includes the details displayed in **show ip ospf neighbor detail**, as shown in the second example below. Software releases 07.6.03 and later include a new command, **show ip ospf neighbor detail**, in addition to the **show ip ospf neighbor** command. Both of these commands are described below.

USING THE CLI

To display OSPF neighbor information, enter the following command at any CLI level:

```
BigIron> 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:

```
BigIron# 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 applies to releases 07.6.03 and later. This parameter displays detailed information about the neighbor routers.

These displays show the following information.

Table 15.3: CLI Display of OSPF Neighbor Information

Field	Description
Port	The port through which the Layer 3 Switch is connected to the neighbor.
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

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.

Table 15.3: CLI Display of OSPF Neighbor Information (Continued)

Field	Description
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	(Service Provider release 09.1.01 and later) The amount of time the Foundry device will wait for a HELLO message from each OSPF neighbor before assuming the neighbor is dead.

USING THE WEB MANAGEMENT INTERFACE

1. Log on to the device using a valid user name and password for read-only or read-write access. The System configuration panel is displayed.
2. Click on the plus sign next to Monitor in the tree view to expand the list of monitoring options.
3. Click on the plus sign next to OSPF in the tree view to expand the list of OSPF option links.
4. Click on the Neighbor link.

Displaying OSPF Interface Information

To display OSPF interface information for the router, use one of the following methods.

USING THE CLI

To display OSPF interface information, enter the following command at any CLI level:

```
BigIron# 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 15.4: 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
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.

USING THE WEB MANAGEMENT INTERFACE

1. Log on to the device using a valid user name and password for read-only or read-write access. The System configuration panel is displayed.
2. Click on the plus sign next to Monitor in the tree view to expand the list of monitoring options.
3. Click on the plus sign next to OSPF in the tree view to expand the list of OSPF option links.
4. Click on the Interface link.

Displaying OSPF Route Information

To display OSPF route information for the router, use one of the following methods.

USING THE CLI

To display OSPF route information, enter the following command at any CLI level:

```
BigIron> 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 15.5: 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.

Table 15.5: CLI Display of OSPF Route Information (Continued)

This Field...	Displays...
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.

USING THE WEB MANAGEMENT INTERFACE

You cannot display the OSPF route table using the Web management interface.

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:

```
BigIron# 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:

```
BigIron# 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 for the router, use one of the following methods.

USING THE CLI

To display external link state information, enter the following command at any CLI level:

```
BigIron> 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 15-62 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 15.6: 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. 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.
Type	The route type, which is always EXTR (external).

USING THE WEB MANAGEMENT INTERFACE

1. Log on to the device using a valid user name and password for read-only or read-write access. The System configuration panel is displayed.
2. Click on the plus sign next to Monitor in the tree view to expand the list of monitoring options.
3. Click on the plus sign next to OSPF in the tree view to expand the list of OSPF option links.
4. Click on the External Link State DB link.

Displaying OSPF Link State Information

To display link state information for the router, use one of the following methods.

USING THE CLI

To display link state information, enter the following command at any CLI level:

```
BigIron> 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 15-62 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.

USING THE WEB MANAGEMENT INTERFACE

1. Log on to the device using a valid user name and password for read-only or read-write access. The System configuration panel is displayed.
2. Click on the plus sign next to Monitor in the tree view to expand the list of monitoring options.
3. Click on the plus sign next to OSPF in the tree view to expand the list of OSPF option links.
4. Click on the Link State DB link.

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:

```
BigIron> 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:

```
BigIron> 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...

USING THE WEB MANAGEMENT INTERFACE

You cannot display the contents of an LSA using the Web management interface.

Displaying OSPF Virtual Neighbor Information

To display OSPF virtual neighbor information for the router, use one of the following methods.

USING THE CLI

To display OSPF virtual neighbor information, enter the following command at any CLI level:

```
BigIron> show ip ospf virtual-neighbor
```

Syntax: show ip ospf virtual-neighbor [<num>]

The <num> parameter displays the table beginning at the specified entry number.

USING THE WEB MANAGEMENT INTERFACE

1. Log on to the device using a valid user name and password for read-only or read-write access. The System configuration panel is displayed.
2. Click on the plus sign next to Monitor in the tree view to expand the list of monitoring options.
3. Click on the plus sign next to OSPF in the tree view to expand the list of OSPF option links.
4. Click on the Virtual Neighbor link.

Displaying OSPF Virtual Link Information

To display OSPF virtual link information for the router, use one of the following methods.

USING THE CLI

To display OSPF virtual link information, enter the following command at any CLI level:

```
BigIron> show ip ospf virtual-link
```

Syntax: show ip ospf virtual-link [<num>]

The <num> parameter displays the table beginning at the specified entry number.

USING THE WEB MANAGEMENT INTERFACE

1. Log on to the device using a valid user name and password for read-only or read-write access. The System configuration panel is displayed.
2. Click on the plus sign next to Monitor in the tree view to expand the list of monitoring options.
3. Click on the plus sign next to OSPF in the tree view to expand the list of OSPF option links.
4. Click on the Virtual Interface link.

Displaying OSPF ABR and ASBR Information

To display OSPF ABR and ASBR information for the router, use one of the following methods.

USING THE CLI

To display OSPF ABR and ASBR information, enter the following command at any CLI level:

```
BigIron> 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.

USING THE WEB MANAGEMENT INTERFACE

1. Log on to the device using a valid user name and password for read-only or read-write access. The System configuration panel is displayed.

2. Click on the plus sign next to Monitor in the tree view to expand the list of monitoring options.
3. Click on the plus sign next to OSPF in the tree view to expand the list of OSPF option links.
4. Click on the ABR ASBR Routers link.

Displaying OSPF Trap Status

To display the state (enabled or disabled) of the OSPF traps, use one of the following methods.

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 15-43.

USING THE CLI

To display the state of each OSPF trap, enter the following command at any CLI level:

```
BigIron> 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

USING THE WEB MANAGEMENT INTERFACE

1. Log on to the device using a valid user name and password for read-only or read-write access. The System configuration panel is displayed.
2. Click on the plus sign next to Configure in the tree view to expand the list of configuration options.
3. Click on the plus sign next to OSPF in the tree view to expand the list of OSPF option links.
4. Click on the Trap link to display the OSPF Trap panel.

Displaying OSPF Graceful Restart Information

Use the **show ip ospf data grace-link-state** and the **show ip ospf neighbor** commands to display information about OSPF graceful restart.

The following is an example of what the **show ip ospf data grace-link-state** command that is displayed during a restart event. The output is blank if the report is requested while the OSPF router is in normal operation.

```
BigIron MG8#show ip ospf data grace-link-state
    Area  Interface  Router ID    Type  Age      Restart-Time  Seq
     0    3/27       12.1.0.14    9     27        120          0x80000001
     0     v31       12.1.0.14    9     27        120          0x80000001
     0     v32       12.1.0.14    9     27        120          0x80000001
     0     v33       12.1.0.14    9     27        120          0x80000001
     0     v34       12.1.0.14    9     27        120          0x80000001
```

The **show ip ospf neighbor** command displays the following information during normal operation:

```
BigIron MG8#show ip ospf neighbor
Port  Address         Pri State      Neigh Address   Neigh ID        Ev Opt Cnt
3/1   30.1.0.5        0   FULL/OTHER 30.1.0.13       30.0.0.13       5  2   0
3/27  25.27.0.8       1   FULL/DR    25.27.0.14      12.1.0.14       20 2   0
v31   21.23.0.5       1   FULL/DR    21.23.0.14      12.1.0.14       15 2   0
v32   22.24.0.5       1   FULL/DR    22.24.0.14      12.1.0.14       15 2   0
v33   23.25.0.5       1   FULL/DR    23.25.0.14      12.1.0.14       15 2   0
v34   24.26.0.5       1   FULL/DR    24.26.0.14      12.1.0.14       15 2   0
```

The **show ip ospf neighbor** command displays the following information during a restart event on a helper router. Note the "<in graceful restart state...>" entry appears only during restart. It does not appear once restart is complete.

```
BigIron MG8#sh ip ospf neigh
Port  Address         Pri State      Neigh Address   Neigh ID        Ev Opt Cnt
3/1   30.1.0.5        0   FULL/OTHER 30.1.0.13       30.0.0.13       5  2   0
3/27  25.27.0.8       1   FULL/DR    25.27.0.14      12.1.0.14       20 2   0
  < in graceful restart state,  helping 1, timer 104 sec >
v31   21.23.0.5       1   FULL/DR    21.23.0.14      12.1.0.14       15 2   0
  < in graceful restart state,  helping 1, timer 104 sec >
v32   22.24.0.5       1   FULL/DR    22.24.0.14      12.1.0.14       15 2   0
  < in graceful restart state,  helping 1, timer 104 sec >
v33   23.25.0.5       1   FULL/DR    23.25.0.14      12.1.0.14       15 2   0
  < in graceful restart state,  helping 1, timer 104 sec >
v34   24.26.0.5       1   FULL/DR    24.26.0.14      12.1.0.14       15 2   0
  < in graceful restart state,  helping 1, timer 104 sec >
```

Clearing OSPF Information from the Foundry Device

The following CLI commands allow you to clear specific kinds of information from the Foundry device's OSPF link state database and OSPF routing table.

NOTE: The following commands are available in Enterprise software release 08.0.00 and later.

The following kinds of OSPF information can be cleared:

- 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:

```
BigIron# 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:

```
BigIron# 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:

```
BigIron# 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:

```
BigIron# 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:

```
BigIron# clear ip ospf
```

Syntax: clear ip ospf [<area-id>]

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:

```
BigIron# clear ip ospf 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.

The <area-id> can be specified in decimal format or in IP address format.

Chapter 16
Configuring BGP4

This chapter provides details on how to configure ***Border Gateway Protocol version 4 (BGP4)*** on Foundry products using the CLI and the Web management interface. BGP4 is supported on the following Foundry products:

- NetIron Internet Backbone router
- BigIron 15000, BigIron 8000, and BigIron 4000 Layer 3 Switches
- NetIron Stackable Layer 3 Switch (must have 32MB RAM and 4MB flash module)

NOTE: BGP4 is not supported on the FastIron II.

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 16-111.

This chapter shows the commands you need in order to configure the Foundry Layer 3 Switch for BGP4. For a detailed list of all CLI commands, including syntax and possible values, see the *Foundry Switch and Router Command Line Interface Reference*.

NOTE: Your Layer 3 Switch’s management module must have 32MB or higher to run BGP4.

NOTE: The TurboIron/8, NetIron Stackable device, and BigIron Layer 3 Switches using basic management modules (not Management 2 or higher) can contain 10,000 routes by default. If you need to increase the capacity of the IP route table for BGP4, see the “Displaying and Modifying System Parameter Default Settings” section of the “Configuring Basic Features” chapter of the *Foundry Switch and Router Installation and Basic Configuration Guide*.

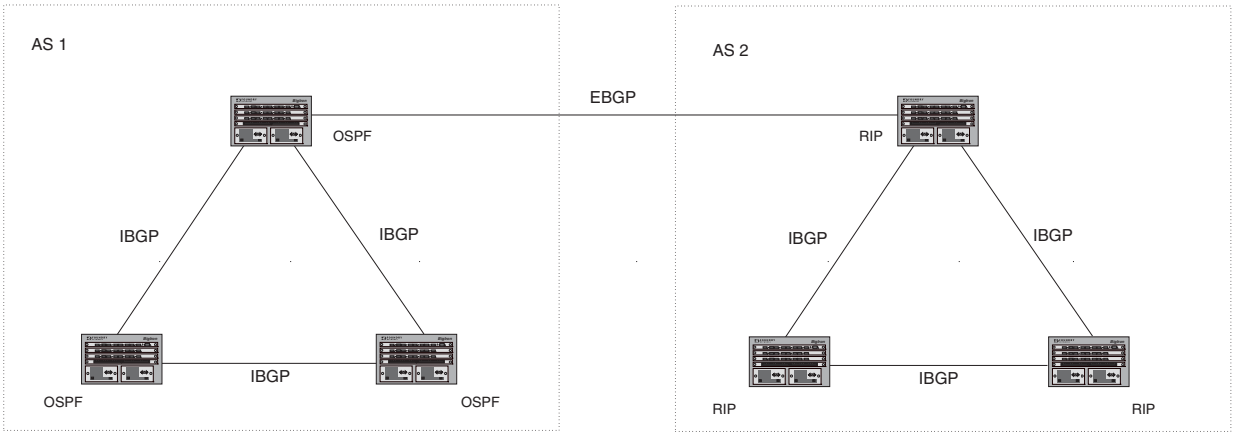
Overview of BGP4

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 16.1 on page 16-2 shows a simple example of two BGP4 ASs. Each AS contains three BGP4 routers. All of the BGP4 routers within an AS communicate using IBGP. BGP4 routers communicate with other ASs using EBGP. Notice that each of the routers also is running an Interior Gateway Protocol (IGP). The routers in AS1 are running OSPF and the routers 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 16.1 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 router 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 router'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 16-4 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 16-27.)

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 16-35.

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 16-40.
   - Beginning in software release 07.5.00, 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***. In software release 07.5.00 and later, deterministic 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.
   - Beginning in software release 07.2.06 and in releases earlier than 07.5.00, 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.)

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. In software release 07.5.00 and later, 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.

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 12-40.

- 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. BGP4 is not functional until you specify the local AS number.

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 12-40. If you change the router ID, all current BGP4 sessions are cleared.

USING THE CLI

NOTE: This procedure shows a command prompt for a BigIron, but the same steps apply to any Foundry Layer 3 Switch that supports BGP4.

```
BigIron> enable
BigIron# configure terminal
BigIron(config)# router bgp
BGP4: Please configure 'local-as' parameter in order to enable BGP4.
BigIron(config-bgp-router)# local-as 10
BigIron(config-bgp-router)# neighbor 209.157.23.99 remote-as 100
BigIron(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.

USING THE WEB MANAGEMENT INTERFACE

1. Log on to the device using a valid user name and password for read-write access. The System configuration panel is displayed.
2. Select the Enable radio button next to BGP.
3. Enter the local AS number in the Local AS field.
4. Click the Apply button to apply the changes to the device's running-config file.
5. 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.

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:

```
BigIron(config-bgp-router)# no router bgp
router bgp mode now disabled. All bgp config data will be lost when writing to
flash!
```

The Web management interface does not display a warning message.

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.

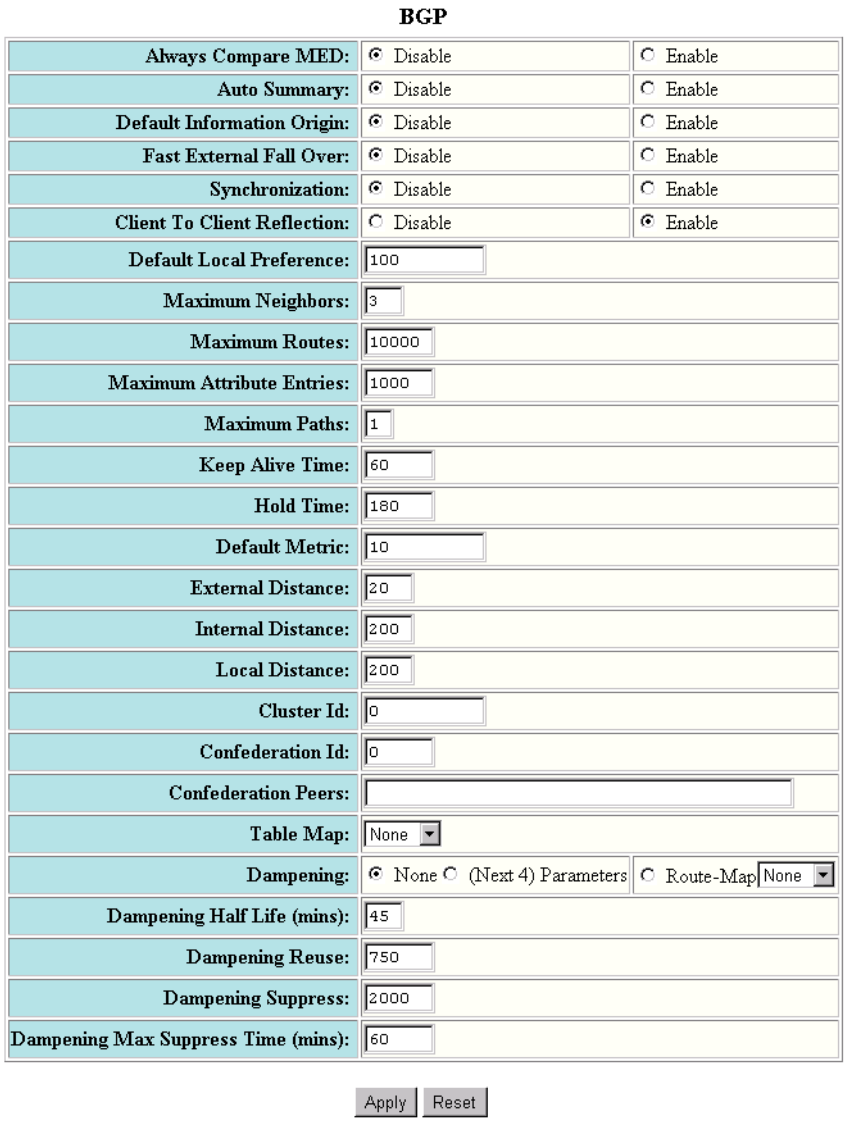
- Optional – Define the router ID. (The same router ID also is used by OSPF.)
- Required – Specify the local AS number.
- Optional – Add a loopback interface for use with neighbors.
- Required – Identify BGP4 neighbors.
- Optional – Change the Keep Alive Time and Hold Time.
- Optional – Change the update 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.

NOTE: When using the Web management interface, you set BGP4 global parameters using the BGP configuration panel, shown in Figure 16.2 on page 16-9. You can access all other parameters using links on the BGP configuration panel or from the Configure->BGP options in the tree view. Select Configure->BGP-General to display the BGP configuration panel.

Figure 16.2 BGP configuration panel



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 16-152.)

- 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 and NetIron Internet Backbone routers 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 16.1 lists the maximum total amount of system memory (DRAM) BGP4 can use in software release 07.1.00. The maximum depends on the total amount of system memory on the device.

Table 16.1: Maximum Memory Usage

Platform	Maximum Memory BGP4 Can Use
Standard management module (Management 1) with 32 MB **Note**: This amount also applies to TurboIron/8 and NetIron Stackable Layer 3 Switches with 32 MB.	7 MB
Management 2 module with 128 MB	62 MB
Management 2, 3, or 4 Module with 256 MB	188 MB
Management 4 module with 512 MB	426 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 16-27.

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:

USING THE CLI

```
BigIron> enable
BigIron# configure terminal
BigIron(config)# router bgp
BGP4: Please configure 'local-as' parameter in order to enable BGP4.
BigIron(config-bgp-router)# local-as 10
BigIron(config-bgp-router)# neighbor 209.157.23.99 remote-as 100
BigIron(config-bgp-router)# write memory
```

USING THE WEB MANAGEMENT INTERFACE

1. Log on to the device using a valid user name and password for read-write access. The System configuration panel is displayed.
2. Select the Enable radio button next to BGP.
3. Enter the local AS number in the Local AS field.
4. Click the Apply button to apply the changes to the device's running-config file.
5. 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.

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 or select the IP->General links from the Configure tree in the Web management interface.

USING THE CLI

To change the router ID, enter a command such as the following:

```
BigIron(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.

USING THE WEB MANAGEMENT INTERFACE

1. Log on to the device using a valid user name and password for read-write access. The System configuration panel is displayed.
2. Click on the plus sign next to Configure in the tree view to expand the list of configuration options.
3. Click on the plus sign next to IP in the tree view to expand the list of IP option links.
4. Click on the General link to display the IP configuration panel.
5. Edit the value in the Router ID field. Specify a valid IP address that is not in use on another device in the network.
6. Click the Apply button to save the change to the device's running-config file.
7. 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.

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, use either of the following methods.

USING THE CLI

To set the local AS number, enter commands such as the following:

```
BigIron(config)# router bgp
BGP4: Please configure 'local-as' parameter in order to enable BGP4.
BigIron(config-bgp-router)# local-as 10
BigIron(config-bgp-router)# write memory
```

Syntax: [no] local-as <num>

The <num> parameter specifies the local AS number.

USING THE WEB MANAGEMENT INTERFACE

1. Log on to the device using a valid user name and password for read-write access. The System configuration panel is displayed.
2. Select the Enable radio button next to BGP.
3. Enter the local AS number in the Local AS field.
4. Click the Apply button to apply the changes to the device's running-config file.
5. 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.

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, use one of the following methods.

USING THE CLI

To add a loopback interface, enter commands such as those shown in the following example:

```
BigIron(config-bgp-router)# exit
BigIron(config)# int loopback 1
BigIron(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 and the TurboIron/8 Layer 3 Switch. The value can be from 1 – 4 on the NetIron Stackable Layer 3 Switch.

USING THE WEB MANAGEMENT INTERFACE

1. Log on to the device using a valid user name and password for read-write access. The System configuration panel is displayed.
2. Select the IP Address link to display a table listing the configured IP addresses.
3. Select the Loop Back link.

   NOTE: If the device already has loopback interfaces, a table listing the interfaces is displayed. Click the Modify button to the right of the row describing an interface to change its configuration, or click the Add Loop Back link to display the Router Loop Back configuration panel.

4. Select the loopback interface number from the Loopback field's pulldown menu. You can select from 1 – 8.
5. Select the status. The interface is enabled by default.
6. Click Add to add the new interface.
7. Click on Configure in the tree view to display the configuration options.
8. Click on IP to display the IP configuration options.
9. Select the Add IP Address link to display the Router IP Address panel.
10. Select the loopback interface from the Port field's pulldown menu. For example, to select loopback interface 1, select "lb1". (If you are configuring a Chassis device, you can have any slot number in the Slot field. Loopback interfaces are not associated with particular slots or physical ports.)
11. Enter the loopback interface's IP address in the IP Address field.
12. Enter the network mask in the Subnet Mask field.
13. Click the Add button to save the change to the device's running-config file.
14. 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.

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 16-23.

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 16-26.

USING THE CLI

To add a BGP4 neighbor with IP address 209.157.22.26, enter the following command:

```
BigIron(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 <portnum> | loopback <num> | pos <portnum> | ve <num>]
[weight <num>]
[local-as <local-as-number>]

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 16-23.

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 16-86.

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 16-58.

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 16-60.

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. For more information, see "Configuring MBGP" on page 17-1.

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 16-21.

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 16-69.

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 16-73.

---

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 16-42. 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 16-147.

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 16-27.

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 16-105.

update-source <ip-addr> | **ethernet** <portnum> | **loopback** <num> | **pos** <portnum> | **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.

See the section "Configuring a Local-AS as a Neighbor" on page 16-18 for the **local-as** <local-as-number>] option.

Configuring a Local-AS as a Neighbor

On the BigIron MG8 and NetIron 40G devices running software release 02.2.01 and later, the Neighbor Local Autonomous System (AS) feature is available. This feature allows a router that is a member of one AS to appear to also be a member of another AS. This feature is useful, for example, if Company A purchses Company B, but Company B does not want to modify it peering configurations.

This feature can be used only for true EBGP peers. When establishing a BGP connection, the router will use the configured neighbor local AS, instead of the system AS number.

For example, if you want a router to use AS 200, instead of 100 when peering with neighbor 11.11.11.2, enter commands such as the following:

```
BigIron MG8(config)#router bgp
BigIron MG8(config-bgp-router)#local-as 100
BigIron MG8(config-bgp-router)#graceful-restart restart-time 30
BigIron MG8(config-bgp-router)#graceful-restart
BigIron MG8(config-bgp-router)#neighbor 11.11.11.2 remote-as 101
BigIron MG8(config-bgp-router)#neighbor 11.11.11.2 local-as 200
```

Syntax: [no] neighbor <ip-address> local-as <local-as-number>

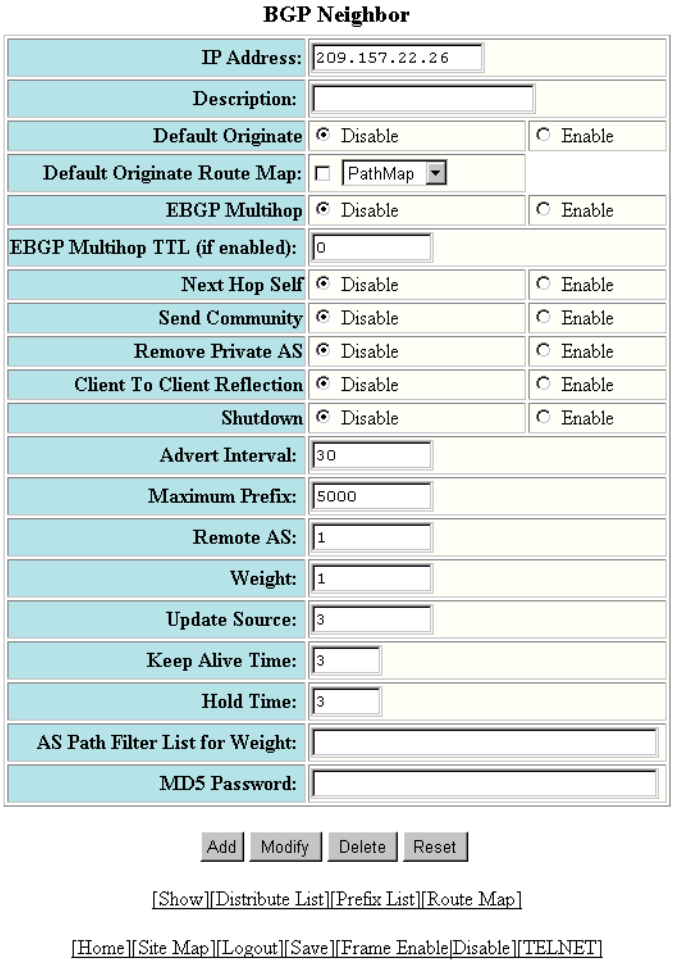
Enter the IP address of the neighbor with which the device will be peering for <ip-address>.

USING THE WEB MANAGEMENT INTERFACE

1. Log on to the device using a valid user name and password for read-write access. The System configuration panel is displayed.
2. Click on the plus sign next to Configure in the tree view to expand the list of configuration options.
3. Click on the plus sign next to BGP in the tree view to expand the list of BGP option links.

4. Click on the Neighbor link to display the BGP Neighbor panel.

NOTE: If the device already has neighbors, a table listing the neighbors is displayed. Click the Modify button to the right of the row describing the neighbor to change its configuration, or click the Add Neighbor link to display the BGP Neighbor configuration panel.



5. Enter the neighbor's IP address in the IP Address field.
6. Enter a description in the Description field.
7. Select Enable next to Default Originate if you want to enable this feature for the neighbor. By default, the Layer 3 Switch does not 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.
8. Select the checkbox next to Default Originate Route Map and select a route map from the pulldown menu if you want to use a route map to control advertisement of default routes.
9. Select Enable next to EBGP Multihop if the neighbor is multiple EBGP hops away.
10. If you enabled EBGP Multihop, enter the TTL for EBGP multihop in the EBGP Multihop TTL field. 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.
11. Select Enable next to Next Hop Self if the router should list itself as the next hop in updates sent to the neighbor. This option is disabled by default.

12. Select Enable next to Send Community if you want to send the community attribute in updates to the neighbor. By default, the router does not send the community attribute.
13. Select Enable next to Remove Private AS if you want 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.
14. Select Enable next to Client To Client Reflection if 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 16-42. This option is disabled by default.
15. Select Enable next to Shutdown if you want to administratively shut 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.
16. Enter the advertisement interval in the Advert Interval field. This parameter 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.
17. Edit the value in the Maximum Prefix field to change the maximum prefix. The maximum prefix is the maximum number of IP network prefixes (routes) that can be learned from the specified neighbor. The default and maximum configurable values depend on the product type:
    - NetIron Internet Backbone router – You can configure a value from 0 – 4294967295. The default is 0 (unlimited).
    - BigIron Layer 3 Switch – Same as for the NetIron 400 and NetIron 800.
    - TurboIron/8 Layer 3 Switch – Same as for the NetIron 400 and NetIron 800.
    - NetIron Stackable Layer 3 Switch – You can configure a value from 100 to the maximum number of BGP4 routes allowed on the Layer 3 Switch. The default is 5000. The maximum value depends on the type of Layer 3 Switch you have and also on whether you have changed the maximum number of routes for the device. See "Changing the Maximum Number of Routes" on page 16-109.
18. Enter the remote AS number in the Remote AS field. The remote AS number is the number of the AS the neighbor is in.
19. Enter the weight you want the Layer 3 Switch to add to routes received from the specified neighbor. BGP4 prefers larger weights over smaller weights. The default weight is 0.
20. Enter the number of an update source loopback interface in the Update Source field. This parameter configures the router to communicate with the neighbor through the loopback address on the specified interface. Using a loopback address for neighbor communication avoids problems that can be caused by unstable router interfaces. Generally, loopback interfaces are used for links to IBGP neighbors, which often are multiple hops away, rather than EBGP neighbors. The loopback interface number can be from 1 – 8. There is no default.
21. Enter a Keep Alive time in the Keep Alive Time field. This parameter overrides the global BGP4 Keep Alive Time configured on the Layer 3 Switch. You can specify from 0 – 65535 seconds. The default is the current global setting.
22. Enter a Hold Time in the Hold Time field. This parameter overrides the global BGP4 Hold Time configured on the Layer 3 Switch. 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 default is the current global setting.

    NOTE: Set the Hold Time to three times the value of the Keep Alive Time. For information about these parameters, see "Changing the Keep Alive Time and Hold Time" on page 16-27.

23. If you specified a weight in the Weight field, enter a list of AS Path filters in the AS Path Filter List for Weight field. 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.

NOTE: By default, if an AS-path 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 AS-path filter must already be configured. See “Filtering AS-Paths” on page 16-60.

24. Enter a password in the MD5 Password field to secure the Layer 3 Switch’s sessions with this neighbor.

NOTE: You must configure the neighbor to use the same password.

25. Click the Add button (if you are adding a new neighbor) or the Modify button (if you are modifying a neighbor that is already configured) to apply the changes to the device’s running-config file.
26. Select the Save link at the bottom of the dialog, then select Yes when prompted to save the configuration change to the startup-config file on the device’s flash memory.

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 are upgrading from a software release earlier than 07.1.14 on a device that is already configured for BGP4, when you save the configuration to the startup-config file, the software automatically converts the command syntax for BGP4 neighbors and peer groups into the new syntax that includes the encryption option. If you display the running-config after reloading with software release 07.1.14 or later, 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 Layer 3 Switch you plan to upgrade. If you need to return to a software release earlier than 07.1.14, the earlier software will not recognize the passwords or authentication keys in their encrypted form and will not be able to convert them back to their clear form.

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.

```
BigIron(config-bgp-router)# local-as 2
BigIron(config-bgp-router)# neighbor xyz peer-group
BigIron(config-bgp-router)# neighbor xyz password abc
BigIron(config-bgp-router)# neighbor 10.10.200.102 peer-group xyz
BigIron(config-bgp-router)# neighbor 10.10.200.102 password test
```


Here is how the commands appear when you display the BGP4 configuration commands:

```
BigIron(config-bgp-router)# 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 in software release 07.1.14 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:

```
BigIron(config)# enable password-display
BigIron(config)# 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, use either of the following methods.

USING THE CLI

To configure a peer group, enter commands such as the following at the BGP configuration level:

```
BigIron(config-bgp-router)# neighbor PeerGroup1 peer-group
BigIron(config-bgp-router)# neighbor PeerGroup1 description “EastCoast Neighbors”
BigIron(config-bgp-router)# neighbor PeerGroup1 remote-as 100
BigIron(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 16-14.

The remaining parameters are the same ones supported for individual neighbors. See "Adding BGP4 Neighbors" on page 16-14.

USING THE WEB MANAGEMENT INTERFACE

You cannot configure peer group parameters using the Web management interface.

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 a neighbor to a peer group, use either of the following methods.

USING THE CLI

To add neighbors to a peer group, enter commands such as the following:

```
BigIron(config-bgp-router)# neighbor 192.168.1.12 peer-group PeerGroup1
BigIron(config-bgp-router)# neighbor 192.168.2.45 peer-group PeerGroup1
BigIron(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.

USING THE WEB MANAGEMENT INTERFACE

You cannot configure peer group parameters using the Web management interface.

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 reestablish 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, use either of the following methods.

USING THE CLI

To shut down a BGP4 neighbor, enter commands such as the following:

```
BigIron(config)# router bgp
BigIron(config-bgp-router)# neighbor 209.157.22.26 shutdown
BigIron(config-bgp-router)# write memory
```

Syntax: [no] neighbor <ip-addr> shutdown

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

USING THE WEB MANAGEMENT INTERFACE

1. Log on to the device using a valid user name and password for read-write access. The System configuration panel is displayed.
2. Click on the plus sign next to Configure in the tree view to expand the list of configuration options.
3. Click on the plus sign next to BGP in the tree view to expand the list of BGP option links.

4. Click on the Neighbor link to display the BGP Neighbor panel.

NOTE: If the device already has neighbors, a table listing the neighbors is displayed. Click the Modify button to the right of the row describing the neighbor to change its configuration, or click the Add Neighbor link to display the BGP Neighbor configuration panel.

5. Enter or modify parameters as needed. For detailed information, see "Adding BGP4 Neighbors" on page 16-14.
6. Select the Enable radio button next to Shutdown.
7. Click the Add button (if you are adding a new neighbor) or the Modify button (if you are modifying a neighbor that is already configured) to apply the changes to the device's running-config file.
8. Select the Save link at the bottom of the dialog, then select Yes when prompted to save the configuration change to the startup-config file on the device's flash memory.

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 16-14.

USING THE CLI

To change the Keep Alive Time to 30 and Hold Time to 90, enter the following command:

```
BigIron(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.

USING THE WEB MANAGEMENT INTERFACE

1. Log on to the device using a valid user name and password for read-write access. The System configuration panel is displayed.
2. Click on the plus sign next to Configure in the tree view to expand the list of configuration options.
3. Click on the plus sign next to BGP in the tree view to expand the list of BGP option links.
4. Click on the General link to display the BGP configuration panel, shown in Figure 16.2 on page 16-9.
5. Edit the number in the Keep Alive Time field. The Keep Alive Time can be 0 – 65535.

6. Edit the number in the Hold Time field. 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.

   NOTE: Generally, you should set the Hold Time to three times the value of the Keep Alive Time.

7. Click the Apply button to apply the changes to the device's running-config file.
8. 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.

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:

```
BigIron(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.

USING THE CLI

To enable fast external fallover, enter the following command:

```
BigIron(config-bgp-router)# fast-external-fallover
```

To disable fast external fallover again, enter the following command:

```
BigIron(config-bgp-router)# no fast-external-fallover
```

Syntax: [no] fast-external-fallover

USING THE WEB MANAGEMENT INTERFACE

1. Log on to the device using a valid user name and password for read-write access. The System configuration panel is displayed.

2. Click on the plus sign next to Configure in the tree view to expand the list of configuration options.
3. Click on the plus sign next to BGP in the tree view to expand the list of BGP option links.
4. Click on the General link to display the BGP configuration panel, shown in Figure 16.2 on page 16-9.
5. Select Disable or Enable next to Fast External Fall Over.
6. Click the Apply button to apply the changes to the device's running-config file.
7. 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.

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 16-3 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 – 8.

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 eight equal paths. You can set the maximum number of paths to a value from 1 – 8. 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 or use the # of Paths field next to Load Sharing on the IP configuration panel of the Web management interface.

USING THE CLI

To change the maximum number of shared paths, enter commands such as the following:

```
BigIron(config)# router bgp
BigIron(config-bgp-router)# maximum-paths 4
BigIron(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 – 8. The default is 1.

USING THE WEB MANAGEMENT INTERFACE

1. Log on to the device using a valid user name and password for read-write access. The System configuration panel is displayed.
2. Click on the plus sign next to Configure in the tree view to expand the list of configuration options.
3. Click on the plus sign next to BGP in the tree view to expand the list of BGP option links.
4. Click on the General link to display the BGP configuration panel, shown in Figure 16.2 on page 16-9.
5. Edit the number in the # of Paths field if needed. You can specify from 1 – 8 paths. The default is 1. You cannot set the maximum number of BGP4 paths to a number higher than the IP load sharing maximum number of paths.
6. Click the Apply button to apply the changes to the device's running-config file.
7. Select the Save link at the bottom of the dialog, then select Yes when prompted to save the configuration change to the startup-config file on the device's flash memory.

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:

```
BigIron(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:

```
BigIron(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:

```
BigIron(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.

USING THE CLI

To configure the Layer 3 Switch to advertise network 209.157.22.0/24, enter the following command:

```
BigIron(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. For more information, see “Configuring MBGP” on page 17-1.

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.

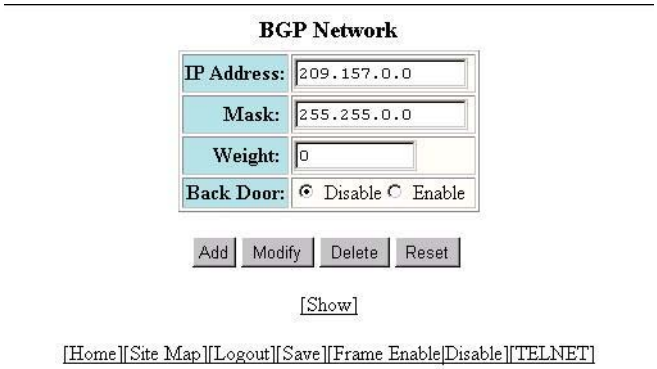
The **weight** <num> parameter specifies a weight to be added to routes to this network.

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.

USING THE WEB MANAGEMENT INTERFACE

1. Log on to the device using a valid user name and password for read-write access. The System configuration panel is displayed.
2. Click on the plus sign next to Configure in the tree view to expand the list of configuration options.
3. Click on the plus sign next to BGP in the tree view to expand the list of BGP option links.
4. Click on the Network link.
   - If the device does not have any BGP networks configured, the BGP Network configuration panel is displayed, as shown in the following example.
   - If a BGP network is already configured and you are adding a new one, click on the Add Network link to display the BGP Network configuration panel, as shown in the following example.

- If you are modifying an existing BGP network, click on the Modify button to the right of the row describing the network to display the BGP Network configuration panel, as shown in the following example.



5. Enter the network address in the IP Address field.
6. Enter the network mask in the Mask field.
7. Optionally enter a weight to be added to routes to this network.
8. If you want to tag the route as a backdoor route, select Enable next to Back Door.
9. Click the Apply button to apply the changes to the device's running-config file.
10. Select the Save link at the bottom of the dialog, then select Yes when prompted to save the configuration change to the startup-config file on the device's flash memory.

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.

USING THE CLI

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:

```
BigIron(config)# route-map set_net permit 1
BigIron(config-routemap set_net)# set community no-export
BigIron(config-routemap set_net)# exit
BigIron(config)# router bgp
BigIron(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 16-73.

USING THE WEB MANAGEMENT INTERFACE

You cannot add a route map to a BGP4 network definition using the Web management interface.

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 16-73. See “How BGP4 Selects a Path for a Route” on page 16-3 for information about the BGP4 algorithm.

To change the default local preference used by the router, use either of the following methods.

USING THE CLI

To change the default local preference to 200, enter the following command:

```
BigIron(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 WEB MANAGEMENT INTERFACE

1. Log on to the device using a valid user name and password for read-write access. The System configuration panel is displayed.
2. Click on the plus sign next to Configure in the tree view to expand the list of configuration options.
3. Click on the plus sign next to BGP in the tree view to expand the list of BGP option links.
4. Click on the General link to display the BGP configuration panel, shown in Figure 16.2 on page 16-9.
5. Change the number in the Default Local Preference field. You can enter a number from 0 – 4294967295.
6. Click the Apply button to apply the changes to the device’s running-config file.
7. 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.

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:

```
BigIron(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.

USING THE CLI

To enable the router to originate and advertise a default BGP4 route, enter the following command:

```
BigIron(config-bgp-router)# default-information-originate
```

Syntax: [no] default-information-originate

USING THE WEB MANAGEMENT INTERFACE

1. Log on to the device using a valid user name and password for read-write access. The System configuration panel is displayed.
2. Click on the plus sign next to Configure in the tree view to expand the list of configuration options.
3. Click on the plus sign next to BGP in the tree view to expand the list of BGP option links.
4. Click on the General link to display the BGP configuration panel, shown in Figure 16.2 on page 16-9.
5. Select Disable or Enable next to Default Information Originate.
6. Click the Apply button to apply the changes to the device's running-config file.
7. 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.

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.

USING THE CLI

To change the default metric to 40, enter the following command:

```
BigIron(config-bgp-router)# default-metric 40
```

Syntax: default-metric <num>

The <num> indicates the metric and can be a value from 0 – 4294967295.

USING THE WEB MANAGEMENT INTERFACE

1. Log on to the device using a valid user name and password for read-write access. The System configuration panel is displayed.
2. Click on the plus sign next to Configure in the tree view to expand the list of configuration options.
3. Click on the plus sign next to BGP in the tree view to expand the list of BGP option links.
4. Click on the General link to display the BGP configuration panel, shown in Figure 16.2 on page 16-9.
5. Change the number in the Default Metric field. You can enter a number from 0 – 4294967295.
6. Click the Apply button to apply the changes to the device's running-config file.

7. 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.

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: In software release 07.2.01 and later, 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.

```
BigIron# 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):

```
BigIron# 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.

```
BigIron# 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.

```
BigIron# 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:

```
BigIron# 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:

```
BigIron# 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:

```
BigIron# 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:

```
BigIron# 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 the feature, use the following CLI method.

USING THE CLI

To enable recursive next-hop lookups, enter the following command at the BGP configuration level of the CLI:

```
BigIron(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 16-3.

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: In software release 05.0.00 and later, 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 in software release 05.2.00, so only directly-connected routes are preferred over static routes when the default administrative distances for the routes are used.

Here are the default administrative distances 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 15-42.
- To change the default administrative distance for RIP, see "Changing the Administrative Distance" on page 13-6.
- To change the default administrative distance for static routes, see "Configuring Static Routes" on page 12-54.

You can change the default EBGP, IBGP, and Local BGP administrative distances using either of the following methods.

USING THE CLI

To change the default administrative distances for EBGP, IBGP, and Local BGP, enter a command such as the following:

```
BigIron(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.

USING THE WEB MANAGEMENT INTERFACE

1. Log on to the device using a valid user name and password for read-write access. The System configuration panel is displayed.
2. Click on the plus sign next to Configure in the tree view to expand the list of configuration options.
3. Click on the plus sign next to BGP in the tree view to expand the list of BGP option links.
4. Click on the General link to display the BGP configuration panel, shown in Figure 16.2 on page 16-9.
5. Change the number in the External Distance field to change the EBGP distance. You can enter a number from 1 – 255.
6. Change the number in the Internal Distance field to change the IBGP distance. You can enter a number from 1 – 255.
7. Change the number in the Local Distance field to change the local distance. You can enter a number from 1 – 255.
8. Click the Apply button to apply the changes to the device's running-config file.
9. 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.

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:

```
BigIron(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:

```
BigIron(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 16-3 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 default in software release 07.5.02. 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:

```
BigIron(config-bgp-router)# compare-routerid
```

Syntax: [no] compare-routerid

For more information, see "How BGP4 Selects a Path for a Route" on page 16-3.

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".

- Beginning in software release 07.5.00, 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***. In software release 07.5.00 and later, deterministic 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.

- Beginning with software release 07.2.06 and in releases earlier than 07.5.00, 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. In software release 07.5.00 and later, 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 for all paths for a route, use either of the following methods:

USING THE CLI

To configure the router to always compare MEDs, enter the following command:

```
BigIron(config-bgp-router)# always-compare-med
```

Syntax: [no] always-compare-med

USING THE WEB MANAGEMENT INTERFACE

1. Log on to the device using a valid user name and password for read-write access. The System configuration panel is displayed.
2. Click on the plus sign next to Configure in the tree view to expand the list of configuration options.
3. Click on the plus sign next to BGP in the tree view to expand the list of BGP option links.
4. Click on the General link to display the BGP configuration panel, shown in Figure 16.2 on page 16-9.
5. Select Disable or Enable next to Always Compare MED.
6. Click the Apply button to apply the changes to the device's running-config file.
7. 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.

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.

USING THE CLI

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:

```
BigIron(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.

USING THE WEB MANAGEMENT INTERFACE

You cannot perform this task using the Web management interface.

Automatically Summarizing Subnet Routes Into Class A, B, or C Networks

The auto summary feature summarizes the routes it redistributes from IGP to BGP4. The router summarizes subnets into their natural class A, B, or C networks. For example, if an AS contains subnets 1.1.0.0, 1.2.0.0, and 1.3.0.0 with the network mask 255.255.0.0, the auto summary feature summarizes the subnets in its advertisements to BGP4 neighbors as 1.0.0.0/8.

The auto summary feature is disabled by default. If you want to enable the feature, use either of the following methods.

NOTE: The auto summary feature summarizes only the routes that are redistributed from IGP into BGP4.

NOTE: The auto summary feature does not summarize networks that use CIDR numbers instead of class A, B, or C numbers. To summarize CIDR networks, use the aggregation feature. See “Aggregating Routes Advertised to BGP4 Neighbors” on page 16-48.

USING THE CLI

To enable auto summary, enter the following command:

```
BigIron(config-bgp-router)# auto-summary
```

To disable auto summary again, enter the following command:

```
BigIron(config-bgp-router)# no auto-summary
```

Syntax: [no] auto-summary

USING THE WEB MANAGEMENT INTERFACE

1. Log on to the device using a valid user name and password for read-write access. The System configuration panel is displayed.
2. Click on the plus sign next to Configure in the tree view to expand the list of configuration options.
3. Click on the plus sign next to BGP in the tree view to expand the list of BGP option links.
4. Click on the General link to display the BGP configuration panel, shown in Figure 16.2 on page 16-9.
5. Select Disable or Enable next to Auto Summary.
6. Click the Apply button to apply the changes to the device’s running-config file.
7. 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.

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.

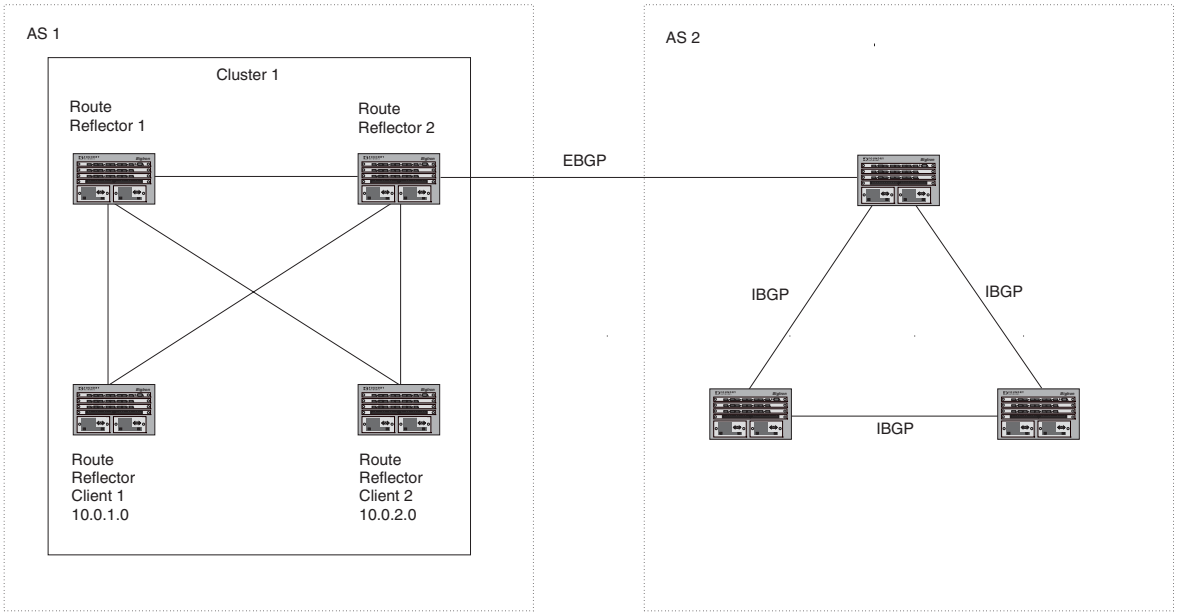
- A ***route reflector*** is an IGP router configured to send BGP route information to all the clients (other BGP4 routers) within the cluster. Route reflection is enabled on all Foundry BGP4 routers by default but does not take effect unless you add route reflector clients to the router.
- A ***route 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 16.3 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 16.3 Example route reflector configuration



Support for RFC 2796

In software release 07.0.10 and higher, route reflection 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 router that originated the route. The route reflector inserts this attribute when reflecting a route to an IBGP neighbor. If a BGP4 router receives an advertisement that contains its own router ID as the ORIGINATOR_ID, the router 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 router discards the advertisement and does not forward it.

Software release 07.0.10 and higher 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.

USING THE CLI

Enter the following commands to configure a Foundry Layer 3 Switch as route reflector 1 in Figure 16.3 on page 16-43. 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.

```
BigIron(config-bgp-router)# cluster-id 1
BigIron(config-bgp-router)# neighbor 10.0.1.0 route-reflector-client
BigIron(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 16-14.

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.

```
BigIron(config-bgp-router)# no client-to-client-reflection
```

Enter the following command to re-enable the feature:

```
BigIron(config-bgp-router)# client-to-client-reflection
```

Syntax: [no] client-to-client-reflection

USING THE WEB MANAGEMENT INTERFACE

1. Log on to the device using a valid user name and password for read-write access. The System configuration panel is displayed.
2. Click on the plus sign next to Configure in the tree view to expand the list of configuration options.
3. Click on the plus sign next to BGP in the tree view to expand the list of BGP option links.
4. Click on the General link to display the BGP configuration panel, shown in Figure 16.2 on page 16-9.
5. If route reflection is not already enabled, select Enable next to Client To Client Reflection.
6. If the autonomous system (AS) the Layer 3 Switch is in will contain more than one route reflector (a route reflector in addition to the Layer 3 Switch), enter a cluster ID in the Cluster ID field. The cluster ID is required to avoid loops in an AS that contains more than one route reflector.
7. Click the Apply button to apply the changes to the device's running-config file.
8. Click on the Neighbor link at the bottom of the BGP configuration panel or under BGP in the Configure section of the tree view.
9. If you have already configured neighbors, a table listing the neighbors is displayed. Click Modify next to the neighbor you want to identify as a route reflector client or select the Add Neighbor link. The BGP configuration panel is displayed.
10. Configure or change other parameters if needed, then identify this neighbor as a route reflector client by selecting Enable next to Client To Client Reflection. See "Adding BGP4 Neighbors" on page 16-14 for information about the other neighbor parameters.
11. Click the Add button to apply the changes to the device's running-config file.
12. 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.

Configuring Confederations

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.

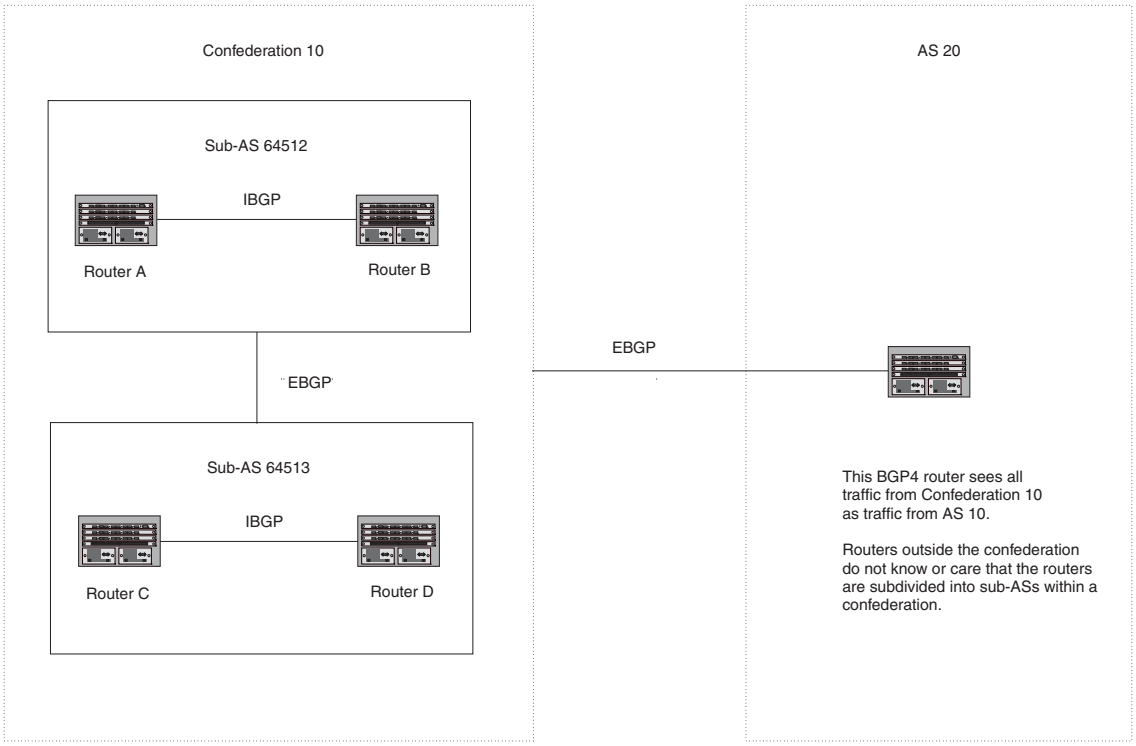
NOTE: Another method for reducing the complexity of an IBGP mesh is to use route reflection. However, if you want to run different Interior Gateway Protocols (IGPs) within an AS, configure a confederation. You can run a separate IGP within each sub-AS.

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 16.4 shows an example of a BGP4 confederation.

Figure 16.4 Example BGP4 confederation



In this example, four routers are configured into two sub-ASs, each containing two of the routers. The sub-ASs are members of confederation 10. Routers within a sub-AS must be fully meshed and communicate using IBGP. In this example, routers A and B use IBGP to communicate. Routers C and D also use IBGP. However, the sub-ASs communicate with one another using EBGP. For example, router A communicates with router C using EBGP. The routers in the confederation communicate with other ASs using EBGP.

Routers in other ASs are unaware that routers A – D are configured in a confederation. In fact, when routers in confederation 10 send traffic to routers in other ASs, the confederation ID is the same as the AS number for the routers in the confederation. Thus, routers in other ASs see traffic from AS 10 and are unaware that the routers 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 routers with the same local AS number are members of the same sub-AS. BGP routers use the local AS number when communicating with other BGP routers within the confederation.
- Configure the confederation ID. The confederation ID is the AS number by which BGP routers outside the confederation know the confederation. Thus, a BGP router outside the confederation is not aware and does not care that your BGP routers are in multiple sub-ASs. BGP routers use the confederation ID when communicating with routers 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 routers within the same sub-AS use IBGP to exchange router information. Routers in different sub-ASs within the confederation use EBGP to exchange router information.

To configure a Layer 3 Switch to be a member of a BGP confederation, use one of the following methods. The procedures show how to implement the example confederation shown in Figure 16.4.

USING THE CLI

To configure four Layer 3 Switches to be a member of confederation 10, consisting of two sub-ASs (64512 and 64513), enter commands such as the following.

Commands for Router A

```
BigIronA(config)# router bgp
BigIronA(config-bgp-router)# local-as 64512
BigIronA(config-bgp-router)# confederation identifier 10
BigIronA(config-bgp-router)# confederation peers 64512 64513
BigIronA(config-bgp-router)# write memory
```

Syntax: local-as <num>

The <num> parameter with the **local-as** command indicates the AS number for the BGP routers 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 routers outside the confederation know the confederation. Thus, a BGP router outside the confederation is not aware and does not care that your BGP routers are in multiple sub-ASs. BGP routers use the confederation ID when communicating with routers 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 routers within the same sub-AS use IBGP to exchange router information. Routers in different sub-ASs within the confederation use EBGP to exchange router information. You can specify a number from 1 – 65535.

Commands for Router B

```
BigIronB(config)# router bgp
BigIronB(config-bgp-router)# local-as 64512
BigIronB(config-bgp-router)# confederation identifier 10
BigIronB(config-bgp-router)# confederation peers 64512 64513
BigIronB(config-bgp-router)# write memory
```


Commands for Router C

```
BigIronC(config)# router bgp
BigIronC(config-bgp-router)# local-as 64513
BigIronC(config-bgp-router)# confederation identifier 10
BigIronC(config-bgp-router)# confederation peers 64512 64513
BigIronC(config-bgp-router)# write memory
```

Commands for Router D

```
BigIronD(config)# router bgp
BigIronD(config-bgp-router)# local-as 64513
BigIronD(config-bgp-router)# confederation identifier 10
BigIronD(config-bgp-router)# confederation peers 64512 64513
BigIronD(config-bgp-router)# write memory
```

USING THE WEB MANAGEMENT INTERFACE

1. Log on to the device using a valid user name and password for read-write access. The System configuration panel is displayed.
2. Click on the plus sign next to Configure in the tree view to expand the list of configuration options.
3. Click on the plus sign next to BGP in the tree view to expand the list of BGP option links.
4. Click on the General link to display the BGP configuration panel, shown in Figure 16.2 on page 16-9.
5. Enter the confederation ID in the Confederation ID field. The confederation ID must be different from the sub-AS numbers. You can specify a number from 1 – 65535.
6. Enter the AS numbers of the peers (sub-ASs) within the confederation in the Confederation Peers field. Separate the AS numbers with spaces. You must specify all the sub-ASs contained in the confederation. All the routers within the same sub-AS use IBGP to exchange router information. Routers in different sub-ASs within the confederation use EBGP to exchange router information. You can specify a number from 1 – 65535.
7. Click the Apply button to apply the changes to the device's running-config file.
8. Select the Save link at the bottom of the dialog, then select Yes when prompted to save the configuration change to the startup-config file on the device's flash 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, use either of the following methods.

USING THE CLI

To aggregate routes for 209.157.22.0, 209.157.23.0, and 209.157.24.0, enter the following command:

```
BigIron(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. For more information, see "Configuring MBGP" on page 17-1.

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.

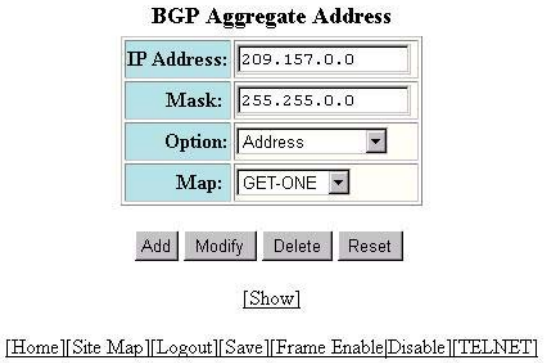
The **advertise-map** <map-name> parameter configures the router to advertise the more specific routes in the specified route map.

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 16-73 for information on defining a route map.

USING THE WEB MANAGEMENT INTERFACE

1. Log on to the device using a valid user name and password for read-write access. The System configuration panel is displayed.
2. Click on the plus sign next to Configure in the tree view to expand the list of configuration options.
3. Click on the plus sign next to BGP in the tree view to expand the list of BGP option links.
4. Click on the Aggregate Address link to display the BGP Aggregate Address configuration panel.
   - If the device does not have any BGP aggregate addresses configured, the BGP Aggregate Address configuration panel is displayed, as shown in the following example.
   - If a BGP aggregate address is already configured and you are adding a new one, click on the Add Aggregate Address link to display the BGP Aggregate Address configuration panel, as shown in the following example.
   - If you are modifying an existing BGP aggregate address, click on the Modify button to the right of the row describing the aggregate address to display the BGP Aggregate Address configuration panel, as shown in the following example.



5. Enter the aggregate address in the IP Address field. 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. Then enter 255.255.0.0 in the Mask field.

6. Enter the mask in the Mask field.
7. Select one of the following options from the Option field's pulldown list:
   - Address – Use this option when you are adding the address. This is the default option.
   - AS Set – This option causes the router to aggregate AS-path information for all the routes in the aggregate address into a single AS-path.
   - Summary Only – This option prevents the router from advertising more specific routes contained within the aggregate route.
   - Suppress Map – This option prevents the more specific routes contained in the specified route map from being advertised.
   - Advertise Map – This option configures the router to advertise the more specific routes in the specified route map.
   - Attribute Map – This option configures the router to set attributes for the aggregate routes based on the specified route map.
8. Optionally select a route map from the Map field's pulldown list.

NOTE: For the Suppress Map, Advertise Map, and Attribute Map options, you must select a route map and the route map must already be defined. See "Defining Route Maps" on page 16-73 for information on defining a route map.

9. Click the Add button to apply the changes to the device's running-config file.
10. 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.

Increasing the Maximum Number of BGP Aggregate Addresses

By default, you can configure up to 20 BGP aggregate addresses on the Foundry device. In Service Provider software release 09.1.00 and higher, you can increase the maximum number of configurable aggregate addresses up to 256. For example, to increase the maximum number to 256, enter the following command at the global CONFIG level of the CLI:

```
NI4802 Router(config)# system-max bgp-aggregate-address 256
```

Syntax: [no] system-max bgp-aggregate-address <number>

The <number> parameter specifies the maximum number of aggregate addresses you can configure on the Foundry device. You can specify a value between 20 – 256.

To restore the default of 20 configurable aggregate addresses, enter the **no** form of this command.

After increasing this limit, you can then configure up to 256 aggregate addresses. For example, to aggregate routes for 192.1.0.0, 192.2.0.0, 192.3.0.0, and so on, enter commands such as the following at the BGP configuration level of the CLI:

```
NI4802 Router(config-bgp-router)# aggregate-address 192.1.0.0 255.255.0.0 summary-
only
NI4802 Router(config-bgp-router)# aggregate-address 192.2.0.0 255.255.0.0 summary-
only
NI4802 Router(config-bgp-router)# aggregate-address 192.3.0.0 255.255.0.0 summary-
only
...
```

Graceful Restart

The Graceful Restart feature provides support for high-availability routing. With this feature enabled, disruptions in forwarding are minimized and route flapping diminished to provide continuous service during times when a router experiences a restart.

NOTE: Graceful restart is available on BigIron MG8 and NetIron 40G devices running software release 02.2.01 and later.

Under normal operation, restarting a BGP router causes the network to be reconfigured. In this situation, routes available through the restarting router are first deleted when the router goes down and are then rediscovered and re-added to the routing tables when the router is back up and running. In a network where routers are restarted regularly, this can degrade performance significantly and limit availability of network resources. BGP graceful restart dampens the network topology changes and limits route flapping by allowing routes to remain available between routers during a restart. BGP Graceful restart operates between a router and its peers and must be configured on both.

A BGP router with graceful restart enabled advertises its graceful restart capability and restart timer to establish peering relationships with other routers. Once the restarting router is restarted, it begins to reestablish BGP connections and receive routing updates from its peers. When the restarting router receives all end-of-RIB markers from its helper neighbors that indicates that it has received all of the BGP route updates, all of the routes are recomputed, and newly computed routes replace the stale routes in the routing table.

During the restarting process, the helper neighbors will continue to use all of the routes learned from the restarting router and mark them as stale for the length of learned restart timer. If the restarting router doesn't come back up within the restart timer, the routes marked stale will be removed.

Configuring BGP Graceful Restart

To configure BGP Graceful Restart, you must enable it on all BGP peers where you want it to operate and set the following timers:

- Restart Timer
- Stale Routes Timer
- Purge Timer

NOTE: After configuring BGP Graceful Restart, you need to reset neighbor session using the **clear ip bgp neighbor** command whether or not the neighbor session is up. This command clears and re-establishes neighbor sessions.

Configuring BGP Graceful Restart on a Router

Use the following command to enable the BGP graceful restart feature on a BigIron MG8 router:

```
BigIron MG8(config)#router bgp
BigIron MG8(config-bgp)#graceful-restart
```

Configuring BGP Graceful Restart Timer

Use the following command to specify the maximum amount of time a router will maintain routes from a restarting router and forward traffic to a restarting router.

```
BigIron MG8(config)#router bgp
BigIron MG8(config-bgp)#graceful-restart
BigIron MG8(config-bgp)#restart-timer 60
```

Syntax: restart-timer <seconds>

The <seconds> variable sets the maximum number of seconds the restarting router will take to restart. Also, the peer routers waits this number of seconds to re-establish BGP connection and to keep using the learned routes from the restarting router. Enter 10 – 3600 seconds. The default value is 120 seconds.

Configuring BGP Graceful Restart Stale Routes Timer

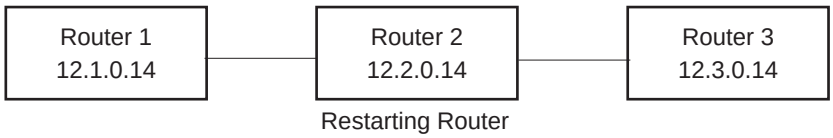
Use the following command to specify the maximum amount of time a helper router will wait for an end-of-RIB message from a restarting router before deleting stale routes learned from that restarting router:

```
BigIron MG8(config-bgp)#stale-routes-time 30
```

Syntax: stale-routes-time <seconds>

The <seconds> variable sets the number of seconds that a helper router will wait for an end-of-RIB (restart complete) message from a restarting router. Enter 10 – 3600 seconds. The default value is 360 seconds.

EXAMPLE:



Router 1

```
BigIron MG8(config)#router bgp
BigIron MG8(config-bgp)#local-as 100
BigIron MG8(config-bgp)#graceful-restart
BigIron MG8(config-bgp)#neighbor 12.2.0.14 remote-as 250
BigIron MG8(config-bgp)#write memory
```

Router 2

```
BigIron MG8(config)#router bgp
BigIron MG8(config-bgp)#local-as 100
BigIron MG8(config-bgp)#graceful-restart
BigIron MG8(config-bgp)#neighbor 12.1.0.14 remote-as 250
BigIron MG8(config-bgp)#neighbor 12.3.0.14 remote-as 250
BigIron MG8(config-bgp)#write memory
```

Router 3

```
BigIron MG8(config)#router bgp
BigIron MG8(config-bgp)#local-as 100
BigIron MG8(config-bgp)#graceful-restart
BigIron MG8(config-bgp)#neighbor 12.2.0.14 remote-as 250
BigIron MG8(config-bgp)#write memory
```


Displaying BGP Graceful Restart information

You can display the BGP Graceful Restart configuration by entering the following command:

```
BigIron MG8#show ip bgp neighbor 11.11.11.2
1 IP Address: 11.11.11.2, Remote AS: 101 (EBGP), RouterID: 101.101.101.1
Local AS: 200
State: ESTABLISHED, Time: 0h18m15s, KeepAliveTime: 60, HoldTime: 180
KeepAliveTimer Expire in 44 seconds, HoldTimer Expire in 167 seconds
RefreshCapability: Received
GracefulRestartCapability: Received
Restart Time 120 sec, Restart bit 0
afi/safi 1/1, Forwarding bit 0
GracefulRestartCapability: Sent
Restart Time 30 sec, Restart bit 0
afi/safi 1/1, Forwarding bit 0
Messages: Open Update KeepAlive Notification Refresh-Req
Sent : 1 5 15 0 0
Received: 1 1 15 0 0
Last Update Time: NLRI Withdraw NLRI Withdraw
Tx: --- --- Rx: --- ---
Last Connection Reset Reason:Unknown
Notification Sent: Unspecified
Notification Received: Unspecified
Neighbor NLRI Negotiation:
Peer Negotiated IPV4 unicast capability
Peer configured for IPV4 unicast Routes
TCP Connection state: ESTABLISHED
TTL check: 0, value: 0, rcvd: 64
Byte Sent: 628, Received: 363
Local host: 11.11.11.1, Local Port: 8190
Remote host: 11.11.11.2, Remote Port: 179
ISentSeq: 2123652 SendNext: 2124281 TotUnAck: 0
TotSent: 629 ReTrans: 1 UnAckSeq: 2124281
IRcvSeq: 2300094 RcvNext: 2300458 SendWnd: 65000
TotalRcv: 364 DupliRcv: 0 RcvWnd: 65000
SendQue: 0 RcvQue: 0 CngstWnd: 1460
```

Syntax: show ip bgp neighbor <address>

Modifying Redistribution Parameters

By default, the router does not redistribute route information between BGP4 and the IP IGPs (RIP and OSPF). You can configure the router to redistribute OSPF routes, RIP routes, directly connected routes, or static routes into BGP4 by using the following methods.

USING THE CLI

To enable redistribution of all OSPF routes and directly attached routes into BGP4, enter the following commands.

```
BigIron(config)# router bgp
BigIron(config-bgp-router)# redistribute ospf
BigIron(config-bgp-router)# redistribute connected
BigIron(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 16-56.

The **rip** parameter indicates that you are redistributing RIP routes into BGP4.

The **static** parameter indicates that you are redistributing static routes into BGP.

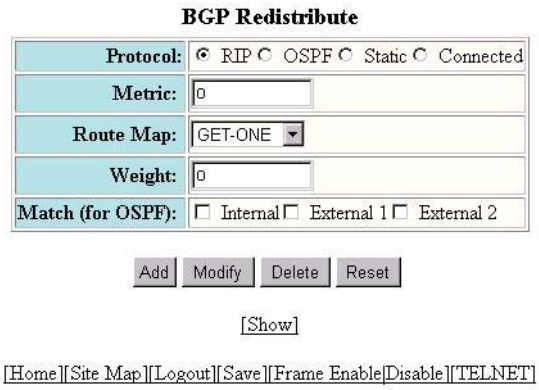
See the following sections for details on redistributing specific routes using the CLI:

- "Redistributing Connected Routes" on page 16-55
- "Redistributing RIP Routes" on page 16-55
- "Redistributing OSPF External Routes" on page 16-56
- "Redistributing Static Routes" on page 16-56

USING THE WEB MANAGEMENT INTERFACE

The following procedure applies to redistributing RIP, OSPF, static, and connected (directly attached) routes.

1. Log on to the device using a valid user name and password for read-write access. The System configuration panel is displayed.
2. Click on the plus sign next to Configure in the tree view to expand the list of configuration options.
3. Click on the plus sign next to BGP in the tree view to expand the list of BGP option links.
4. Click on the Redistribute link to display the BGP Redistribute configuration panel.
   - If the device does not have any BGP redistribution parameters configured, the BGP Redistribute configuration panel is displayed, as shown in the following example.
   - If BGP redistribution parameters are already configured and you are adding new ones, click on the Add Redistribute link to display the BGP Redistribute configuration panel, as shown in the following example.
   - If you are modifying existing BGP redistribution parameters, click on the Modify button to the right of the row describing the redistribution parameters to display the BGP Redistribute configuration panel, as shown in the following example.



5. Select the source of the routes you want to redistribute into BGP4. You can select RIP, OSPF, Static, or Connected (directly attached) routes.
6. Optionally enter a metric for the redistributed routes in the Metric field. You can specify a value from 0 – 4294967295. The default is 0.
7. Optionally select a route map from the Map field's pulldown list.

NOTE: The route map must already be defined. See “Defining Route Maps” on page 16-73 for information on defining a route map.

8. Optionally enter a weight for the redistributed routes in the Weight field. You can specify a value from 0 – 65535. The default is 0.
9. For OSPF routes, select one of the following to specify the types of OSPF routes to be redistributed into BGP4:
   - Internal
   - External 1
   - External 2

NOTE: If you do not indicate the route type, then OSPF routes will be redistributed internally.

10. Click the Add button to apply the changes to the device’s running-config file.
11. 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.

Redistributing Connected Routes

To configure BGP4 to redistribute directly connected routes, enter the following command:

```
BigIron(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 router. See “Defining Route Maps” on page 16-73 for information about defining route maps.

Redistributing RIP Routes

USING THE CLI

To configure BGP4 to redistribute RIP routes and add a metric of 10 to the redistributed routes, enter the following command:

```
BigIron(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. 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 router. See “Defining Route Maps” on page 16-73 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:

```
BigIron(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. 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 OSPF route to the BGP4 route table.

NOTE: The route map you specify must already be configured on the router. See “Defining Route Maps” on page 16-73 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:

```
BigIron(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. 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 static route to the BGP4 route table.

NOTE: The route map you specify must already be configured on the router. See “Defining Route Maps” on page 16-73 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.

USING THE CLI

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:

```
BigIron(config-bgp-router)# no readvertise
```


Syntax: [no] readvertise

To re-enable re-advertisement, enter the following command:

```
BigIron(config-bgp-router)# readvertise
```

USING THE WEB MANAGEMENT INTERFACE

You cannot configure this parameter using the Web management interface.

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.

USING THE CLI

To enable the Layer 3 Switch to redistribute BGP4 routes into OSPF and RIP, enter the following command:

```
BigIron(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:

```
BigIron(config-bgp-router)# no bgp-redistribute-internal
```

USING THE WEB MANAGEMENT INTERFACE

You cannot configure this parameter using the Web management interface.

Redistributing Filter Rebinding

In previous releases, if you modified OSPF redistribution filters in the Foundry device's configuration, you then had to remove and reapply the **redistribution rip** and **redistribution static** statements in order for OSPF to start redistributing routes based on the filters.

Starting in release 08.0.00, a new CLI command rebinds the redistribution filters, so you no longer have to remove and reapply the **redistribution rip** and **redistribution static** statements in the Foundry device's configuration.

For example, if the Foundry device's configuration contained the following statements:

```
permit redistribute 1 static address 130.126.0.12 255.255.255.255
permit redistribute 2 static address 128.174.201.0 255.255.255.128
permit redistribute 3 rip
deny redistribute 64 all
redistribution rip
redistribution static
```

and you then added the following filter statement:

```
permit redistribute 4 static address 192.17.220.0 255.255.254.0
```

you then had to remove the **redistribution rip** and **redistribution static** statements and then re-enter them in order for OSPF to redistribute routes based on the filter statements.

In this release, instead of removing and reapplying the **redistribution rip** and **redistribution static** statements, you can enter the following commands after modifying OSPF redistribution filters:

```
BigIron(config)# router ospf
BigIron(config-ospf-router)# redistribution rebind
```

Syntax: redistribution rebind

After you enter the **redistribution rebind** command, OSPF redistributes routes based on all of the filter statements in the configuration. Note that the **redistribution rebind** command is not stored in the Foundry device's configuration.

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. See "Access Control List" on page 6-1.

To define an IP address filter, use either of the following methods.

USING THE CLI

To define an IP address filter to deny routes to 209.157.0.0, enter the following command:

```
BigIron(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.

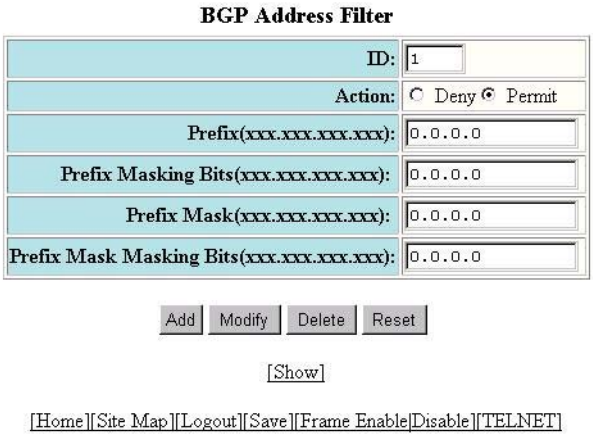
If you prefer to specify the wildcard (mask value) in Classless Interdomain Routing (CIDR) format, you can enter a forward slash after the IP address, then enter the number of significant bits in the mask. For example, you can enter the CIDR equivalent of "209.157.22.26 0.0.0.255" as "209.157.22.26/24". 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.

USING THE WEB MANAGEMENT INTERFACE

1. Log on to the device using a valid user name and password for read-write access. The System configuration panel is displayed.
2. Click on the plus sign next to Configure in the tree view to expand the list of configuration options.
3. Click on the plus sign next to BGP in the tree view to expand the list of BGP option links.
4. Click on the Address Filter link to display the BGP Address Filter panel.
   - If the device does not have any BGP address filters configured, the BGP Address Filter configuration panel is displayed, as shown in the following example.
   - If BGP address filters are already configured and you are adding a new one, click on the Add Address Filter link to display the BGP Address Filter configuration panel, as shown in the following example.
   - If you are modifying an existing BGP address filter, click on the Modify button to the right of the row describing the filter to display the BGP Address Filter configuration panel, as shown in the following example.



5. Enter the filter ID in the ID field. You can specify a number from 1 – 100.

6. Select the action you want the Layer 3 Switch to perform if the filter is true:
   - If you select Deny, the router denies the route from entering the BGP4 table if the filter match is true.
   - If you select Permit, the router permits the route into the BGP4 table if the filter match is true.
7. Enter the network prefix in the Prefix field. If you specify “any”, all networks match the filter.
8. Enter the prefix masking bits in the Prefix Masking Bits field. The prefix masking bits indicate the bits in the prefix that the filter compares. The filter disregards the bits for which the mask contains zeros.
9. Enter the mask in the Prefix Mask field. If you specify “any”, all masks match the filter.
10. Enter the masking bits for the network mask in the Prefix Mask Masking Bits field.
11. Click the Add button to apply the changes to the device’s running-config file.
12. 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.

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 an AS-path filter, use either of the following methods.

USING THE CLI

To define AS-path filter 4 to permit AS 2500, enter the following command:

```
BigIron(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.

USING THE WEB MANAGEMENT INTERFACE

1. Log on to the device using a valid user name and password for read-write access. The System configuration dialog is displayed.
2. Click on the plus sign next to Configure in the tree view to expand the list of configuration options.
3. Click on the plus sign next to BGP in the tree view to expand the list of BGP option links.
4. Click on the AS Path Filter link to display the BGP AS Path Filter panel.
   - If the device does not have any BGP AS-path filters configured, the BGP AS Path Filter configuration panel is displayed, as shown in the following example.
   - If BGP AS-path filters are already configured and you are adding a new one, click on the Add AS Path Filter link to display the BGP AS Path Filter configuration panel, as shown in the following example.
   - If you are modifying an existing BGP AS-path filter, click on the Modify button to the right of the row describing the filter to display the BGP AS Path Filter configuration panel, as shown in the following example.

5. Enter the filter ID in the ID field. You can specify a number from 1 – 100.
6. Select the action you want the Layer 3 Switch to perform if the filter is true:
   - If you select Deny, the router denies the route from entering the BGP4 table if the filter match is true.
   - If you select Permit, the router permits the route into the BGP4 table if the filter match is true.
7. Enter the AS path you want to filter in the Regular Expression field. As indicated by the field's title, you can use regular expressions for the AS path. See "Using Regular Expressions" on page 16-63.
8. Click the Add button to apply the changes to the device's running-config file.
9. 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.

Defining an AS-Path ACL

To configure an AS-path ACL, use either of the following methods.

USING THE CLI

To configure an AS-path list that uses ACL 1, enter a command such as the following:

```
BigIron(config)# ip as-path access-list 1 permit 100
BigIron(config)# router bgp
BigIron(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.

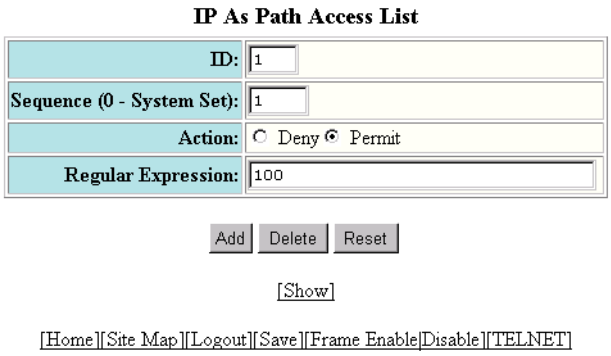
The **deny | permit** parameter specifies the action the software takes if a route's AS-path list matches a match statement in this ACL. To configure the AS-path match statements in a route map, use the **match as-path** command. See "Matching Based on AS-Path ACL" on page 16-78.

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 16-63.

The **neighbor** command uses the **filter-list** parameter to apply the AS-path ACL to the neighbor. See "Adding BGP4 Neighbors" on page 16-14.

USING THE WEB MANAGEMENT INTERFACE

1. Log on to the device using a valid user name and password for read-write access. The System configuration panel is displayed.
2. Click on the plus sign next to Configure in the tree view to display the list of configuration options.
3. Click on the plus sign next to IP to display the list of IP configuration options.
4. Select the AS Path Access List link.
   - If the device does not have any AS Path ACLs, the IP AS Path Access List panel is displayed, as shown in the following example.
   - If an AS Path ACL is already configured and you are adding a new one, click on the Add AS Path Access List link to display the IP AS Path Access List panel, as shown in the following example.



NOTE: You cannot modify an AS Path ACL. Instead, you can delete and then re-add the ACL. To delete an ACL, click on the Delete button to the right of the row describing the ACL, then click on the Add AS Path Access List link.

5. Edit the ACL ID in the ID field, if needed. You can enter a number from 1 – 199.
6. Edit the number in the sequence number in the Sequence field, if you want to override the automatically generated 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.

7. Select the action you want the software to perform if a route's AS path list matches this ACL entry. You can select Deny or Permit.
8. Enter a regular expression to specify the AS path information you want to permit or deny to routes that match this ACL entry. You can enter a specific AS number or use a regular expression. For the regular expression syntax, see "Using Regular Expressions" on page 16-63.
9. Click the Add button to save the change to the device's running-config file.
10. Repeat steps 6 – 9 for each entry in the ACL. To create another AS Path ACL, go to step 5.
11. 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.

NOTE: You cannot apply the AS path ACLs to a neighbor using the Web management interface. You must use the CLI. The AS Path Filter List for Weight field in the BGP Neighbor panel of the Web management interface is not used for AS path filtering, but is instead used for changing a route's weight based on the AS path list.

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:

```
BigIron(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:

```
BigIron(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 16.2 on page 16-63 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 16.2: 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*

Table 16.2: BGP4 Special Characters for Regular Expressions (Continued)

Character	Operation
+	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_
[ ]	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.

Table 16.2: BGP4 Special Characters for Regular Expressions (Continued)

Character	Operation
|	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 16.2 on page 16-63, 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 "*****".

```
BigIron(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 "****".

```
BigIron(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

USING THE CLI

To define filter 3 to permit routes that have the NO_ADVERTISE community, enter the following command:

```
BigIron(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 Confederations" on page 16-45.

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 Confederations" on page 16-45.

USING THE WEB MANAGEMENT INTERFACE

1. Log on to the device using a valid user name and password for read-write access. The System configuration panel is displayed.
2. Click on the plus sign next to Configure in the tree view to expand the list of configuration options.
3. Click on the plus sign next to BGP in the tree view to expand the list of BGP option links.
4. Click on the Community Filter link to display the BGP Community Filter panel.

   NOTE: If the device already has community filters, a table listing the filters is displayed. Click the Modify button to the right of the row describing a filter to change its configuration, or select the Add Community Filter link to display the BGP Community Filter panel.

5. Enter the filter's position in the ID filter. The ID is 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.

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.

6. Select the action for the filter. You can select Deny or Permit:
   - If you select Deny, the router denies the route from entering the BGP4 table if the filter match is true.
   - If you select Permit, the router permits the route into the BGP4 table if the filter match is true.
7. Specify a well-known community you want the Layer 3 Switch to apply to a route when the route matches the filter by selecting from the following:
   - Internet – Filters 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.
   - Local AS – Filters for routes with the well-known community "LOCAL_AS". A route in this community should not be advertised outside the sub-AS. This community type applies to confederations.
   - No Advertise – Filters for routes with the well-known community "NO_ADVERTISE". A route in this community should not be advertised to any BGP4 neighbors.
   - No Export – 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.

NOTE: If you want to filter on a private (administrator-defined) community, do not select one of these. Instead, enter the community number in the Community List field.

8. Specify private communities by entering the community names in the Community List field. Enter the names in the following format <num>:<num>. You can use commas or spaces to separate the names.
9. Click the Add button (if you are adding a new filter) or the Modify button (if you are changing a filter) to apply the changes to the device's running-config file.
10. Select the Save link at the bottom of the dialog, then select Yes when prompted to save the configuration change to the startup-config file on the device's flash memory.

Defining a Community ACL

To configure a community ACL, use either of the following methods.

USING THE CLI

To configure community ACL 1, enter a command such as the following:

```
BigIron(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 16-80 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 16-80.

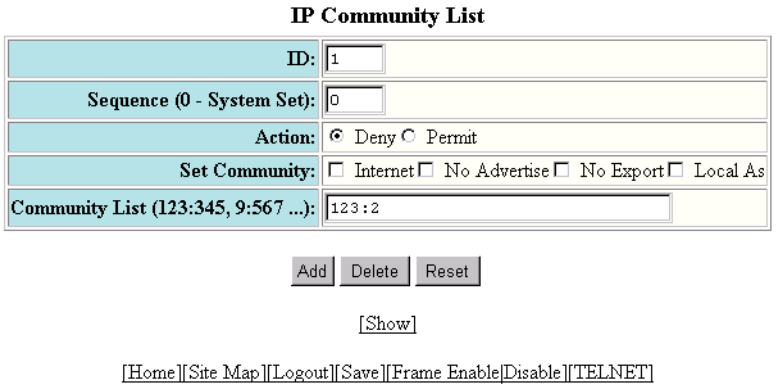
The <community-num> parameter specifies the community type or community number. This parameter can have the following values:

- <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 16-63. You can specify a regular expression only in an extended community ACL.

USING THE WEB MANAGEMENT INTERFACE

1. Log on to the device using a valid user name and password for read-write access. The System configuration panel is displayed.
2. Click on the plus sign next to Configure in the tree view to display the list of configuration options.
3. Click on the plus sign next to IP to display the list of IP configuration options.
4. Select the Community Access List link.
   - If the device does not have any community ACLs, the IP Community List panel is displayed, as shown in the following example.
   - If a community ACL is already configured and you are adding a new one, click on the Add Community Access List link to display the IP Community List panel, as shown in the following example.



NOTE: You cannot modify a community ACL. Instead, you can delete and then re-add the ACL. To delete an ACL, click on the Delete button to the right of the row describing the ACL, then click on the Add Community List link.

5. Edit the ACL ID in the ID field, if needed. You can enter a number from 1 – 199.
6. Edit the number in the sequence number in the Sequence field, if you want to override the automatically generated 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 ascending sequence order.

7. Select the action you want the software to perform if a route's community list matches this ACL entry.
8. Select the community type by clicking on the checkbox to the left of the description, or enter the community numbers in the Community List field.
9. Click the Add button to save the change to the device's running-config file.
10. Repeat steps 6 – 9 for each entry in the ACL. To create another community ACL, go to step 5.
11. 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.

NOTE: You cannot apply the community list ACLs to a neighbor using the Web management interface. You must use the CLI.

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.

USING THE CLI

To configure an IP prefix list and apply it to a neighbor, enter commands such as the following:

```
BigIron(config)# ip prefix-list Routesfor20 permit 20.20.0.0/24
BigIron(config)# router bgp
BigIron(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

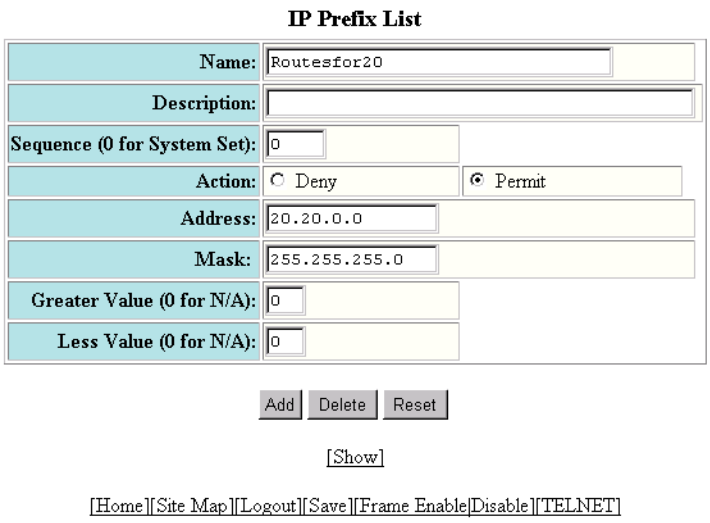
If you do not specify **ge** <ge-value> or **le** <le-value>, the prefix list matches only on the exact network prefix you specify with the <network-addr>/<mask-bits> parameter.

For the syntax of the **neighbor** command shown in the example above, see “Adding BGP4 Neighbors” on page 16-14.

USING THE WEB MANAGEMENT INTERFACE

To configure an IP Prefix List, use the following procedure:

1. Log on to the device using a valid user name and password for read-write access. The System configuration panel is displayed.
2. Click on the plus sign next to Configure in the tree view to display the list of configuration options.
3. Click on the plus sign next to IP to display the list of IP configuration options.
4. Select the Prefix List link.
   - If the device does not have any prefix list ACLs, the IP Prefix List panel is displayed, as shown in the following example.
   - If a prefix list ACL is already configured and you are adding a new one, click on the Add IP Prefix List link to display the IP Prefix List panel, as shown in the following example.



NOTE: You cannot modify an IP prefix list ACL. Instead, you can delete and then re-add the ACL. To delete an ACL, click on the Delete button to the right of the row describing the ACL, then click on the Add IP Prefix List link.

5. Edit a name in the Name field.
6. Enter a description in the Description field.
7. Edit the number in the sequence number in the Sequence field, if you want to override the automatically generated 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.
8. Select the action you want the software to perform if a neighbor’s route is in this prefix list.

9. Enter the IP prefix by entering a network address and subnet mask in the Address and Mask fields.

NOTE: If you do not specify a Greater Value or Less Value, this prefix list entry matches only on the exact network prefix you specified with the values in the Address and Mask fields.

10. Enter a number from 1 – 32 in the Greater Value field if you want the prefix list to match on prefixes that are more specific than the one you entered in the Address and Mask fields, in addition to matching on the prefix in those fields. The value you enter here specifies the minimum number of mask bits in the network mask. For example, if you enter 24 in the example panel shown above, the prefix list matches on all network numbers that are equal to or more specific than 20.20.0.0. Thus 20.20.1.0 and higher also match the prefix list.
11. Enter a number from 1 – 32 in the Less Value field if you want the prefix list to match on prefixes that are less specific than the one you entered in the Address and Mask fields, in addition to matching on the prefix in those fields.
12. Click the Add button to save the change to the device's running-config file.
13. Repeat steps 5 – 12 for each IP prefix list entry.
14. 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.

To apply the IP Prefix List to a neighbor, use the following procedure:

1. In the tree view, click on the plus sign next to BGP under Configure to display the list of BGP configuration options.
2. Select the Neighbor link to display the BGP Neighbor panel.
3. Select the Prefix List link to display the BGP Neighbor Prefix List panel, as shown in the following example.

4. Select the neighbor's IP address from the IP Address field's pulldown menu.

NOTE: The address appears in this menu only if you have already configured the neighbor information on the Layer 3 Switch.

5. Select the direction to which you are applying the prefix list by clicking next to In or Out.
   - In – The prefix list applies to routes received from the neighbor.
   - Out – The prefix list applies to routes destined to be sent to the neighbor.
6. Enter the prefix list name or ID in the Prefix List Name field.
7. Click the Add button to save the change to the device's running-config file.
8. 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.

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.

USING THE CLI

To configure a distribute list that uses ACL 1, enter a command such as the following:

```
BigIron(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 in software release 06.5.00. 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.

USING THE WEB MANAGEMENT INTERFACE

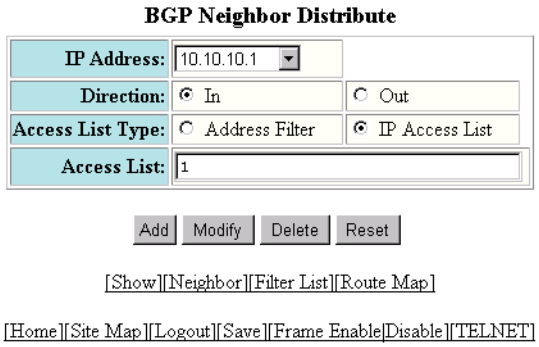
1. Log on to the device using a valid user name and password for read-write access. The System configuration panel is displayed.
2. Click on the plus sign next to Configure in the tree view to expand the list of configuration options.
3. Click on the plus sign next to BGP in the tree view to expand the list of BGP option links.
4. Click on the Neighbor link to display the BGP Neighbor panel.

   NOTE: If the device already has neighbors, a table listing the neighbors is displayed. Click the Modify button to the right of the row describing the neighbor to change its configuration, or click the Add Neighbor link to display the BGP Neighbor configuration panel.

   ---

5. If you are adding a new neighbor or you need to change additional parameters, see the complete procedure in “Adding BGP4 Neighbors” on page 16-14.

6. Select the Distribute List link at the bottom of the panel to display the BGP Neighbor Distribute panel, as shown in the following example.



7. Select the neighbor's IP address from the IP Address field's pulldown menu.

NOTE: The address appears in this menu only if you have already configured the neighbor information on the Layer 3 Switch.

8. Select the direction to which you are applying the distribute list by clicking next to In or Out.
   - In – The distribute list applies to routes received from the neighbor.
   - Out – The distribute list applies to routes destined to be sent to the neighbor.
9. Select the type of distribute list you are applying. You can select one of the following:
   - Address Filter – a BGP4 address filter.
   - IP Access List – an ACL.
10. Enter the address filter or ACL name or ID in the Access List field.
11. Click the Add button to save the change to the device's running-config file.
12. 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.

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

USING THE CLI

To add instance 1 of a route map named "GET_ONE" with a permit action, enter the following command.

```
BigIron(config)# route-map GET_ONE permit 1
BigIron(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 16-76 and “Setting Parameters in the Routes” on page 16-82.

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.

```
BigIron(config)# no route-map Map1
```

This command deletes a route map named “Map1”. All entries in the route map are deleted.

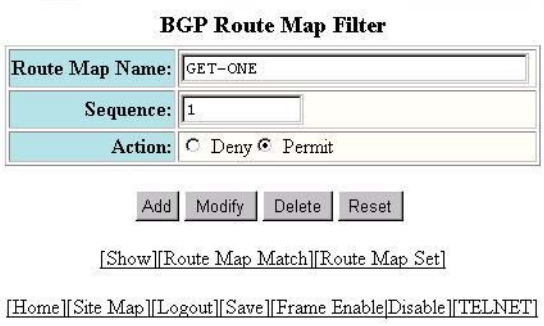
To delete a specific instance of a route map without deleting the rest of the route map, enter a command such as the following:

```
BigIron(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.

USING THE WEB MANAGEMENT INTERFACE

1. Log on to the device using a valid user name and password for read-write access. The System configuration panel is displayed.
2. Click on the plus sign next to Configure in the tree view to expand the list of configuration options.
3. Click on the plus sign next to BGP in the tree view to expand the list of BGP option links.
4. Click on the Route Map Filter link to display the BGP Route Map Filter panel.
   - If the device does not have any BGP route map filters configured, the BGP Route Map Filter configuration panel is displayed, as shown in the following example.
   - If BGP route map filters are already configured and you are adding a new one, click on the Route Map Filter link to display the BGP Route Map Filter configuration panel, as shown in the following example.
   - If you are modifying an existing BGP route map filter, click on the Modify button to the right of the row describing the filter to display the BGP Route Map Filter configuration panel, as shown in the following example.



5. Enter the name of the route map in the Route Map Name field.

6. Enter the sequence (instance) number in the Sequence field. The Layer 3 Switch applies the instances in ascending numerical order. Once an instance comparison results in a “true” evaluation, the Layer 3 Switch stops applying instances and applies the match and set statements you configure for the instance. See “Specifying the Match Conditions” on page 16-76 and “Setting Parameters in the Routes” on page 16-82.
7. Select the action you want the Layer 3 Switch to perform if the comparison results in a “true” value:
   - If you select Deny, the Layer 3 Switch does not advertise or learn the route.
   - If you select Permit, the Layer 3 Switch applies the match and set statements associated with this route map instance.
8. Click the Add button to apply the changes to the device’s running-config file.
9. 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.

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.

```
BigIron(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 16-61.

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 16-58.
- To configure an AS-path filter or AS-path ACL, see “Filtering AS-Paths” on page 16-60.
- To configure a community filter or community ACL, see “Filtering Communities” on page 16-65.

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 "Access Control List" on page 6-1. To configure an IP prefix list, use the **ip prefix-list** command. See "Defining IP Prefix Lists" on page 16-69.

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.

USING THE WEB MANAGEMENT INTERFACE

NOTE: To simplify testing and configuration, you can specify an option and then choose whether to activate it. To activate an option, select the checkbox in front of the option's field. Leave the checkbox unselected to leave the option inactive.

1. If you have just added the route map and the map is displayed in the BGP Route Map Filter panel, go to step 7. Otherwise, go to step 2.
2. Log on to the device using a valid user name and password for read-write access. The System configuration panel is displayed.
3. Click on the plus sign next to Configure in the tree view to expand the list of configuration options.
4. Click on the plus sign next to BGP in the tree view to expand the list of BGP option links.
5. Click on the Route Map Filter link to display a table listing the configured BGP route maps.
6. Click Modify next to the route map you want to configure to display the map in the BGP Route Map Filter panel.
7. Select the Route Map Match link at the bottom of the panel to display the BGP Route Map Match panel.
8. Select the sequence (instance) from the Route Map Name Sequence field's pulldown list. The Layer 3 Switch applies the instances in ascending numerical order and stops after the first match.
9. For OSPF routes, select the one of the following route types—Internal, External1, or External2.
10. Select the type of ACL or filter you are adding as a match condition. You can select more than one ACL or filter type. In this example, select AS Path Access List.

    NOTE: The AS-path, community, and address filters must already be configured.

    NOTE: The Layer 3 Switch does not actively support both filters and ACLs at the same time. Use one method or the other.

    NOTE: IP prefix lists and neighbor distribute lists provide separate means for the same type of filtering. To simplify configuration, Foundry Networks recommends you use one method or the other but do not mix them.

11. Enter the filter or ACL numbers or names in the entry fields next to the filter or ACL types you selected.

12. Optionally enter an IP address against which you want to compare the route updates' next-hop attribute. Enter the address in the Next Hop List field. Also select the checkbox in front of the field.
13. Optionally enter a tag value against which you want to compare the updates in the Tag List field. Also select the checkbox in front of the field.
14. Optionally enter a MED (metric) value against which you want to compare the route updates in the Metric field. Also select the checkbox in front of the field.
15. Click the Apply button to apply the changes to the device's running-config file.
16. 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.

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 match statements for a route map that match based on AS-path information, use either of the following methods.

USING THE CLI

To construct a route map that matches based on AS-path ACL 1, enter the following commands:

```
BigIron(config)# route-map PathMap permit 1
BigIron(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 16-61.

USING THE WEB MANAGEMENT INTERFACE

1. Log on to the device using a valid user name and password for read-write access. The System configuration panel is displayed.
2. Click on the plus sign next to Configure in the tree view to expand the list of configuration options.
3. Click on the plus sign next to BGP in the tree view to expand the list of BGP option links.
4. Click on the Route Map Filter link to display the BGP Route Map Filter panel.
   - If the device does not have any BGP route map filters configured, the BGP Route Map Filter configuration panel is displayed, as shown in the following example.
   - If BGP route map filters are already configured and you are adding a new one, click on the Route Map Filter link to display the BGP Route Map Filter configuration panel, as shown in the following example.
   - If you are modifying an existing BGP route map filter, click on the Modify button to the right of the row describing the filter to display the BGP Route Map Filter configuration panel, as shown in the following

example.

BGP Route Map Filter

Route Map Name:	PathMap
Sequence:	1
Action:	Deny Permit

Add Modify Delete Reset

[Show][Route Map Match][Route Map Set]

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

5. Enter the name of the route map in the Route Map Name field.
6. Enter the sequence (instance) number in the Sequence field. The Layer 3 Switch applies the instances in ascending numerical order. Once an instance comparison results in a “true” evaluation, the Layer 3 Switch stops applying instances and applies the match and set statements you configure for the instance.
7. Select the action you want the Layer 3 Switch to perform if the comparison results in a “true” value:
   - If you select Deny, the Layer 3 Switch does not advertise or learn the route.
   - If you select Permit, the Layer 3 Switch applies the match and set statements associated with this route map instance.
8. Click the Add button to apply the changes to the device’s running-config file.
9. Select the Route Map Match link at the bottom of the panel to display the BGP Route Map Match panel, as shown in the following example.

BGP Route Map Match

Route Map Name.Sequence:	PathMap.1
Route Type:	Internal External1 External2
As Path Filter:	
As Path Access List:	1
Community Filter:	
Community Access List:	
Address Filter:	
IP Addr Access (Name and/or Number) List:	
IP Addr Prefix Name List:	
Next Hop List:	
IP Next Hop Access (Name and/or Number) List:	
IP Next Hop Prefix Name List:	
Tag List:	
Metric:	0

Apply Reset

[Show][Route Map Route][Route Map Set]

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

10. Select the type of ACL or filter you are adding as a match condition. You can select more than one ACL or filter type. In this example, select AS Path Access List.

    NOTE: IP prefix lists and neighbor distribute lists provide separate means for the same type of filtering. To simplify configuration, Foundry Networks recommends you use one method or the other but do not mix them.

11. Next to each type of ACL or filter you selected, enter the ACL or filter name or ID. In this example, AS-path ACL 1 is specified.
12. Click the Apply button to save the change to the device's running-config file.
13. 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.

Matching Based on Community ACL

To construct match statements for a route map that match based on community information, use either of the following methods.

USING THE CLI

To construct a route map that matches based on community ACL 1, enter the following commands:

```
BigIron(config)# ip community-list 1 permit 123:2
BigIron(config)# route-map CommMap permit 1
BigIron(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 16-67.

USING THE WEB MANAGEMENT INTERFACE

Use the procedure in "Matching Based on AS-Path ACL" on page 16-78, but select Community Access List instead of AS Path Access List.

Matching Based on Destination Network

To construct match statements for a route map that match based on destination network, use either of the following methods. You can use the results of an IP ACL or an IP prefix list as the match condition.

USING THE CLI

To construct a route map that matches based on destination network, enter commands such as the following:

```
BigIron(config)# route-map NetMap permit 1
BigIron(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 "Access Control List" on page 6-1.

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 16-69.

USING THE WEB MANAGEMENT INTERFACE

Use the procedure in "Matching Based on AS-Path ACL" on page 16-78, but select IP Addr Access (Name and/or Number) List instead of AS Path Access List.

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.

USING THE CLI

To construct a route map that matches based on the next-hop router, enter commands such as the following:

```
BigIron(config)# route-map HopMap permit 1
BigIron(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 "Access Control List" on page 6-1.

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 16-69.

USING THE WEB MANAGEMENT INTERFACE

Use the procedure in "Matching Based on AS-Path ACL" on page 16-78, but select IP Next Hop Access (Name and/or Number) List instead of AS Path Access List.

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:

```
BigIron(config)# access-list 10 permit 192.168.6.0 0.0.0.255
BigIron(config)# route-map bgp1 permit 1
BigIron(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

Previous software releases enable you to match routes based on the presence of a community name or number in a route. Software release 07.5.00 extends this support by enabling you 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.

```
BigIron(config)# ip community-list standard std_1 permit 12:34 no-export
BigIron(config)# route-map bgp2 permit 1
BigIron(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.

```
BigIron(config)# ip community-list standard std_2 permit 23:45 56:78
BigIron(config)# route-map bgp3 permit 1
BigIron(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.

```
BigIron(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 | pos <portnum>] |
[ip [default] next hop <ip-addr>]
[ip next-hop peer-address] |
[level level-1 | level-1-2 | level-2] |
[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 16-94.

The **[default] interface null0 | pos** <portnum> 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. Alternatively, you can send the traffic to a POS interface. 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). See "Access Control List" on page 6-1.

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). See "Hardware-Based Policy-Based Routing" on page 7-1.

The **ip next-hop peer-address** parameter sets the BGP4 next hop for a route to the specified neighbor address.

The **level level-1 | level-1-2 | level-2** parameter sets the IS-IS level. See the "Configuring IS-IS" chapter in the *Foundry NetIron Service Provider Configuration and Management Guide*.

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 16-85.

The **weight** <num> parameter sets the weight for the route. You can specify a weight value from 0 – 4294967295.

USING THE WEB MANAGEMENT INTERFACE

NOTE: To simplify testing and configuration, you can specify an option and then choose whether to activate it. To activate an option, select the checkbox in front of the option's field. Leave the checkbox unselected to leave the option inactive.

1. If you have just added the route map and the map is displayed in the BGP Route Map Filter panel, go to step 7. Otherwise, go to step 2.
2. Log on to the device using a valid user name and password for read-write access. The System configuration panel is displayed.
3. Click on the plus sign next to Configure in the tree view to expand the list of configuration options.
4. Click on the plus sign next to BGP in the tree view to expand the list of BGP option links.
5. Click on the Route Map Filter link to display a table listing the configured BGP route maps.
6. Click Modify next to the route map you want to configure to display the map in the BGP Route Map Filter panel.
7. Select the Route Map Set link at the bottom of the panel to display the BGP Route Map Set panel.
8. Select the sequence (instance) from the Route Map Name Sequence field's pulldown list.
9. Optionally select the origin. You can select IGP or Incomplete. Also select the checkbox in front of the field.
10. Optionally enter AS numbers to append to the AS path. Also select the checkbox in front of the field.
11. Optionally select Auto Tag. The Layer 3 Switch calculates and sets an automatic tag value for the route.
12. If you did not select Auto Tag and you instead want to set the tag value manually, enter a tag value from 0 – 4294967295 in the Tag field. Also select the checkbox in front of the field.
13. Optionally select the community type and also select the checkbox.
14. For a private community, enter the community number in the Number field. You can enter more than one community. Use commas or spaces to separate the community names.
15. Select Additive of you want the Set statement to add the specified community.
16. Optionally enter a local preference in the Local Preference and also select the checkbox in front of the field. The default local preference is 100. You can set the preference to a value from 0 – 4294967295.
17. Optionally enter a metric (MED) in the Metric field and also select the checkbox in front of the field. The default MED value is 0. You can set the preference to a value from 0 – 4294967295.
18. Optionally enter the Next Hop IP address in the NextHop field and also select the checkbox in front of the field.
19. Optionally enter a weight in the Weight field and also select the checkbox in front of the field. You can specify a weight value from 0 – 4294967295.
20. Click the Apply button to apply the changes to the device's running-config file.
21. 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.

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:

```
BigIron(config)# access-list 1 permit 192.168.9.0 0.0.0.255
BigIron(config)# route-map bgp4 permit 1
BigIron(config-routemap bgp4)# match ip address 1
BigIron(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:

```
BigIron(config)# route-map bgp5 permit 1
BigIron(config-routemap bgp5)# match ip address 1
BigIron(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:

```
BigIron(config)# ip community-list standard std_3 permit 12:99 12:86
BigIron(config)# route-map bgp6 permit 1
BigIron(config-routemap bgp6)# match ip address 1
BigIron(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.

USING THE CLI

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.

```
BigIron(config)# route-map TAG_IP permit 1
BigIron(config-routemap TAG_IP)# match address-filters 11
BigIron(config-routemap TAG_IP)# set tag 100
BigIron(config-routemap TAG_IP)# router bgp
BigIron(config-bgp-router)# table-map TAG_IP
```

USING THE WEB MANAGEMENT INTERFACE

1. Log on to the device using a valid user name and password for read-write access. The System configuration panel is displayed.
2. Use the Web management procedures in “Defining Route Maps” on page 16-73 to create the route map.
3. Click on the plus sign next to Configure in the tree view to expand the list of configuration options.
4. Click on the plus sign next to BGP in the tree view to expand the list of BGP option links.
5. Click on the General link to display the BGP configuration panel.
6. Select the route map name from the Table Map field’s pulldown menu.
7. Click the Apply button to apply the changes to the device’s running-config file.
8. 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.

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:

```
BigIron(config)# ip prefix-list Routesfrom1234 deny 20.20.0.0/24
BigIron(config)# ip prefix-list Routesfrom1234 permit 0.0.0.0/0 le 32
BigIron(config)# router bgp
BigIron(config-bgp-router)# neighbor 1.2.3.4 prefix-list Routesfrom1234 in
BigIron(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:

```
BigIron# 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:

```
BigIron# 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.

```
BigIron# 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:

```
BigIron# 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

Advertising an IBGP Next Hop as a null0 Route as a Defense Against DDoS Attacks

In a distributed denial of service (DDoS) attack, a substantial amount of traffic may be directed at a targeted host or network. In this situation, you can create a static route to forward traffic destined to the targeted host or network to the null0 interface. Traffic forwarded to the null0 interface is dropped in hardware.

Starting in release 08.0.00, you can also use IBGP to advertise the null0 route to other routers in the network, so that the other routers drop the traffic for the targeted host or network.

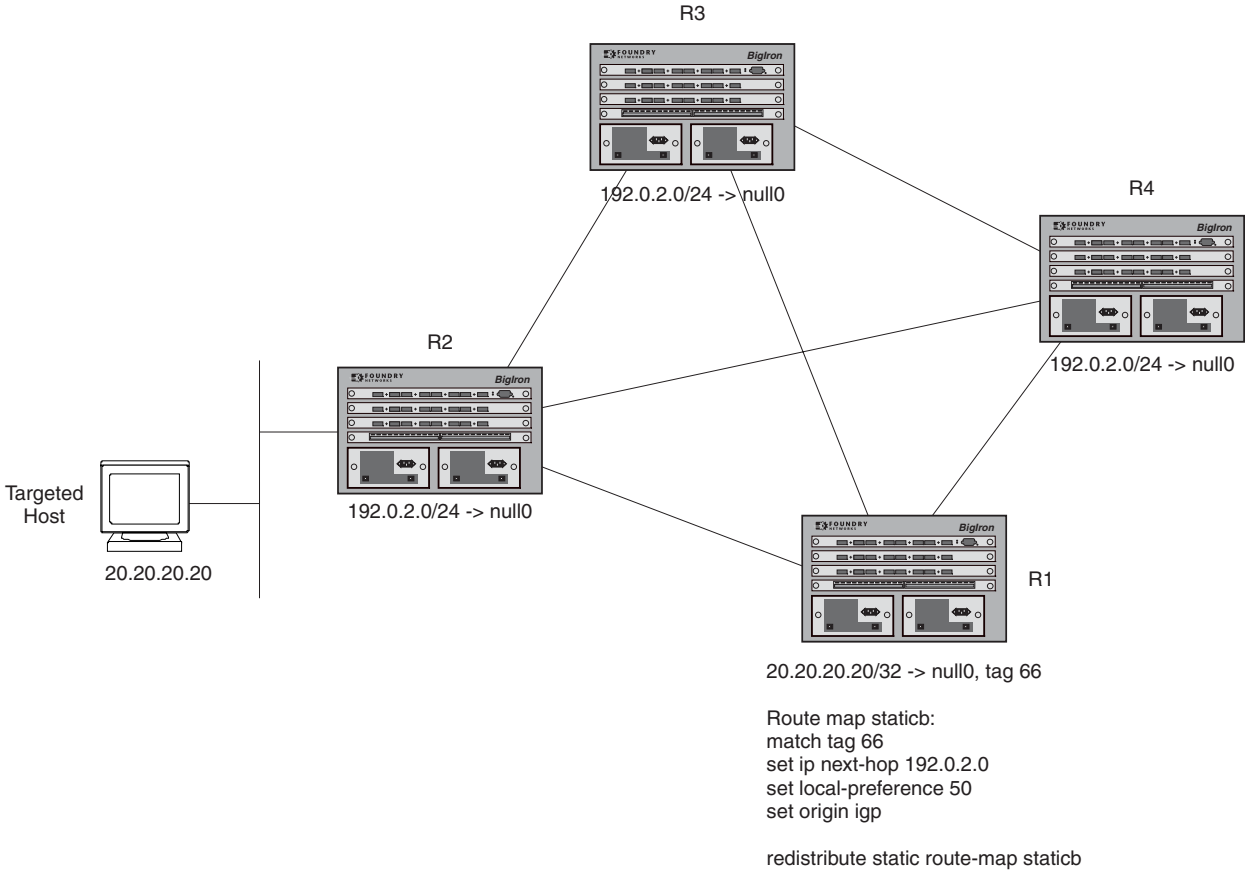
To do this, you perform the following tasks:

1. Configure a null0 route on the other routers in the network.
2. On one of the Foundry routers, configure a static route for the targeted host or network. The static route is configured with a tag so that only specific routes will be redistributed via IBGP, rather than all static routes.
3. Create a route map that matches the tag, sets a local preference value that causes the route to be the preferred route, and configures the origin as IBGP.
4. Redistribute the routes matching the route map via IBGP.

NOTE: This feature applies only to Layer 3 network prefixes. All services for the targeted network prefix are filtered out. To use Layer 4 information as criteria for discarding packets, use an extended ACL.

For example, Figure 16.5 illustrates a configuration where host 20.20.20.20 is targeted in a DDoS attack.

Figure 16.5 Host targeted in a DDoS attack



In this configuration, routers R2, R3, and R4 have a static route that sends packets for 192.0.2.0/24 to the null0 interface. On a Foundry device, this static route is configured with the following command:


```
BigIron(config)# ip route 192.0.2.0 255.255.255.0 null0
```

On router R1, a static route is configured for the targeted host 20.20.20.20, specifies the null0 interface as the route's path, and sets a tag value of 66.

```
BigIron(config)# ip route 20.20.20.20 255.255.255.255 null0 tag 66
```

Syntax: ip route <ip-address> <subnet-mask> null0 tag <number>

On R1, a route map is configured that matches the tag 66, sets the next-hop IP address for traffic that matches tag 66 to 192.0.2.0, which is the address configured for the static null0 route on the other routers in the network. The route map also specifies a local preference value that causes the route to be the preferred route, and configures the origin as IBGP. The commands to configure this route map are as follows:

```
BigIron(config)# route-map staticb
BigIron(config-routemap staticb)# match tag 66
BigIron(config-routemap staticb)# set ip next-hop 192.0.2.0
BigIron(config-routemap staticb)# set local-preference 50
BigIron(config-routemap staticb)# set origin igp
BigIron(config-routemap staticb)# exit
```

The following commands cause static routes that match the staticb route map to be redistributed via IBGP:

```
BigIron(config)# router bgp
BigIron(config-bgp-router)# redistribute static route-map staticb
BigIron(config-bgp-router)# exit
```

Once the route update is processed, 20.20.20.20, the address of the target under attack, is installed as a null0 route on all of the routers in the network. Traffic to the targeted host is discarded.

The output of the **show ip route** command has been enhanced to display the null0 routes. For example:

```
BigIron# show ip route
Total number of IP routes: 7
B:BGP D:Connected  R:RIP  S:Static  O:OSPF *:Candidate default
        Destination      NetMask            Gateway         Port       Cost   Type
        0.0.0.0          0.0.0.0            3.3.3.253        v50       1      S
1       3.3.3.0          255.255.255.0      0.0.0.0          v50       1      D
2       4.4.4.0          255.255.255.0      0.0.0.0         4/37       1      D
3       20.20.20.0       255.255.255.0      255.255.255.255 drop               B
4       192.100.5.0      255.255.255.0      255.255.255.255 drop               B
5       192.168.0.1      255.255.255.255    255.255.255.255 drop        1      S
6       192.168.101.0    255.255.255.0      0.0.0.0          lb1        1      D
```

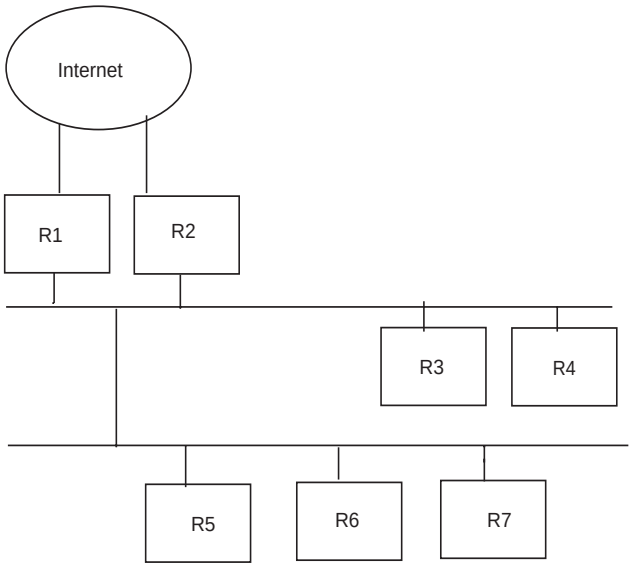
Resolving BGP Next Hop Using Null0 Routing

In previous releases, null0 routes were treated as invalid routes for BGP next hop resolution. Beginning with software release 02.2.01 on the BigIron MG8 and NetIron 40G, BGP can use 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 16.6 shows a topology for a null0 routing application example.

Figure 16.6 SAMPLE Null0 Routing Application



The following steps configure a null0 routing application for stopping denial of service attacks from remote host(s) on the internet.

Configuration Steps

1. Select Router 6, 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 Router 6, redistribute the static routes into BGP, using route-map name in the **redistribute static route-map** command.
6. On Router 1, the router facing the internet, configure a null0 route matching the next-hop address in the route-map

   ```
   BigIron MG8(config)#ip route 199.199.1.1/32 null0.
   ```

   Syntax: ip route <ip-address> <subnet-mask> null0
7. Repeat step 3 for all routers interfacing with the internet (edge corporate routers). In this case, Router 2 has the same null0 route as Router 1.
8. On Router 6, 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

Router 6

The following configuration defines specific prefixes to filter:

```
BigIron MG8(config)#ip route 110.0.0.40/29 ethernet 3/7 tag 50
BigIron MG8(config)#ip route 115.0.0.192/27 ethernet 3/7 tag 50
BigIron MG8(config)#ip route 120.014.0/23 ethernet 3/7 tag 50
```

The following configuration redistributes routes into BGP:

```
BigIron MG8(config)#router bgp
```



```
BigIron MG8(config-bgp-router)#local-as 100
BigIron MG8(config-bgp-router)#neighbor <router1_int_ip address> remote-as 100
BigIron MG8(config-bgp-router)#neighbor <router2_int_ip address> remote-as 100
BigIron MG8(config-bgp-router)#neighbor <router3_int_ip address> remote-as 100
BigIron MG8(config-bgp-router)#neighbor <router4_int_ip address> remote-as 100
BigIron MG8(config-bgp-router)#neighbor <router5_int_ip address> remote-as 100
BigIron MG8(config-bgp-router)#neighbor <router7_int_ip address> remote-as 100
BigIron MG8(config-bgp-router)#redistribute static route-map blockuser
BigIron MG8(config-bgp-router)#exit
```

The following configuration defines the specific next hop address and sets the local preference to preferred:

```
BigIron MG8(config)#route-map blockuser permit 10
BigIron MG8(config-routemap blockuser)#match tag 50
BigIron MG8(config-routemap blockuser)#set ip next-hop 199.199.1.1
BigIron MG8(config-routemap blockuser)#set local-preference 1000000
BigIron MG8(config-routemap blockuser)#set origin igp
BigIron MG8(config-routemap blockuser)#exit
```

Router 1

The following configuration defines the null0 route to the specific next hop address. The next hop address 199.199.1.1 points to 128.178.1.101, which gets blocked:

```
BigIron MG8(config)# ip route 199.199.1.1/32 null0
```

```
BigIron MG8(config)#router bgp
local-as 100
BigIron MG8(config-bgp-router)#neighbor <router2_int_ip address> remote-as 100
BigIron MG8(config-bgp-router)#neighbor <router3_int_ip address> remote-as 100
BigIron MG8(config-bgp-router)#neighbor <router4_int_ip address> remote-as 100
BigIron MG8(config-bgp-router)#neighbor <router5_int_ip address> remote-as 100
BigIron MG8(config-bgp-router)#neighbor <router6_int_ip address> remote-as 100
BigIron MG8(config-bgp-router)#neighbor <router7_int_ip address> remote-as 100
```

Router 2

The following configuration defines a null0 route to the specific next hop address. The next hop address 199.199.1.1 points to 128.178.1.101, which gets blocked:

```
BigIron MG8(config)#ip route 199.199.1.1/32 null0
BigIron MG8(config)#router bgp
BigIron MG8(config-bgp-router)#local-as 100
BigIron MG8(config-bgp-router)#neighbor <router1_int_ip address> remote-as 100
BigIron MG8(config-bgp-router)#neighbor <router3_int_ip address> remote-as 100
BigIron MG8(config-bgp-router)#neighbor <router4_int_ip address> remote-as 100
BigIron MG8(config-bgp-router)#neighbor <router5_int_ip address> remote-as 100
BigIron MG8(config-bgp-router)#neighbor <router6_int_ip address> remote-as 100
BigIron MG8(config-bgp-router)#neighbor <router7_int_ip address> remote-as 100
```

Displaying Null0 Route Information

After configuring the null0 application, use the **show ip route static** command to display information about the route.

Router 6

The following is the **show ip route static** output for Router 6:

```
BigIron MG8#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
BigIron MG8#
```

Router 1 and 2

The following is the **show ip route static** for Router 1 and Router 2:

```
BigIron MG8# 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            192.168.0.1/32     DIRECT          drop        1/1        S
BigIron MG8#
```

Router 6

The following is the **show ip bgp route** for Router-6 which shows the router's routing table.

```
Router-6#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      192.168.0.1     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     192.168.0.1     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      192.168.0.1     1        1000000 32768  BL
         AS_PATH: ..                  .                     .          .      .
```


Router 1 and 2

The **show ip route** output for Router 1 and Router 2 shows "drop" under the Port column for the network prefixes you configured with null0 routing:

```
BigIron MG8#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     192.168.0.1/32     DIRECT          drop        1/1        S
BigIron MG8#
```

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 configure route flap dampening globally, use either of the following methods.

USING THE CLI

To enable route flap dampening using the default values, enter the following command:

```
BigIron(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.

```
BigIron(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 THE WEB MANAGEMENT INTERFACE

1. Log on to the device using a valid user name and password for read-write access. The System configuration panel is displayed.

2. Click on the plus sign next to Configure in the tree view to expand the list of configuration options.
3. Click on the plus sign next to BGP in the tree view to expand the list of BGP option links.
4. Click on the General link to display the BGP configuration panel.
5. Select (Next 4) Parameters next to Dampening, to indicate that you want to enable dampening. This selection also ensures that when you click Apply, the interface applies changes you make to the dampening parameters in the following four fields.
6. Edit the value in the Dampening Half Life field if you want to change the half life. The half like 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.
7. Edit the value in the Dampening Reuse field if you want to change the dampening reuse parameter. The dampening 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").
8. Edit the value in the Dampening Suppress field if you want to change the dampening suppress parameter. The dampening 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").
9. Edit the value in the Dampening Max Suppress Time field if you want to change the maximum suppression parameter. The maximum suppression 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.
10. Click the Apply button to apply the changes to the device's running-config file.
11. 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.

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 route flap dampening for specific routes, use one of the following methods.

USING THE CLI

To configure address filters and a route map for dampening specific routes, enter commands such as the following:

```
BigIron(config)# router bgp
BigIron(config-bgp-router)# address-filter 9 permit 209.157.22.0 255.255.255.0
255.255.255.0 255.255.255.0
BigIron(config-bgp-router)# address-filter 10 permit 209.157.23.0 255.255.255.0
255.255.255.0 255.255.255.0
BigIron(config-bgp-router)# exit
BigIron(config)# route-map DAMPENING_MAP permit 9
BigIron(config-routemap DAMPENING_MAP)# match address-filters 9
BigIron(config-routemap DAMPENING_MAP)# set dampening 10 200 2500 40
BigIron(config-routemap DAMPENING_MAP)# exit
BigIron(config)# route-map DAMPENING_MAP permit 10
BigIron(config-routemap DAMPENING_MAP)# match address-filters 10
BigIron(config-routemap DAMPENING_MAP)# set dampening 20 200 2500 60
BigIron(config-routemap DAMPENING_MAP)# router bgp
BigIron(config-bgp-router)# dampening route-map DAMPENING_MAP
```

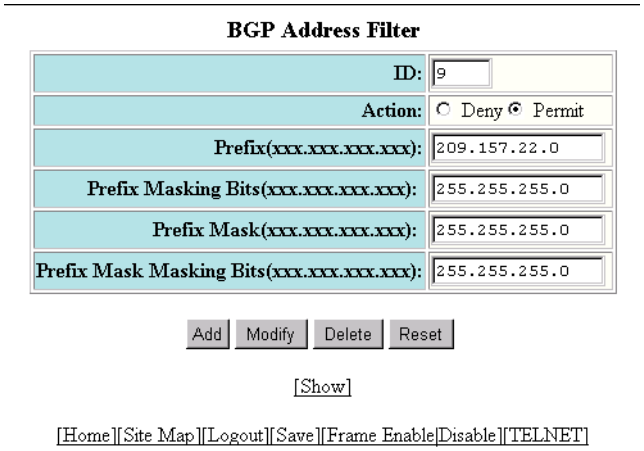
The **address-filter** commands in this example configure two BGP4 address filters, for networks 209.157.22.0 and 209.157.23.0. The first route-map command creates an entry in a route map called “DAMPENING_MAP”. Within this entry of the route map, the **match** command matches based on address filter 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 THE WEB MANAGEMENT INTERFACE

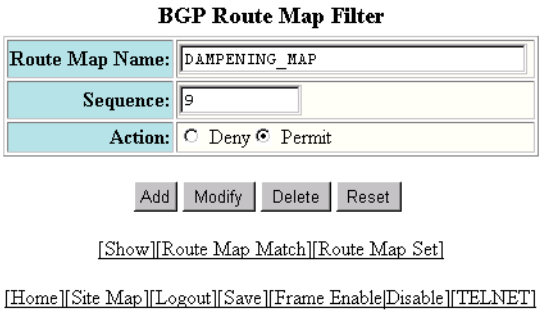
1. Log on to the device using a valid user name and password for read-write access. The System configuration panel is displayed.
2. Click on the plus sign next to Configure in the tree view to expand the list of configuration options.
3. Click on the plus sign next to BGP in the tree view to expand the list of BGP option links.
4. Click on the Address Filter link to display the BGP Address Filter panel.
   - If the device does not have any BGP address filters configured, the BGP Address Filter configuration panel is displayed, as shown in the following example.
   - If BGP address filters are already configured and you are adding a new one, click on the Add Address Filter link to display the BGP Address Filter configuration panel, as shown in the following example.
   - If you are modifying an existing BGP address filter, click on the Modify button to the right of the row describing the filter to display the BGP Address Filter configuration panel, as shown in the following

example.



5. Enter the filter ID in the ID field. You can specify a number from 1 – 100.
6. Select the action you want the Layer 3 Switch to perform if the filter is true:
   - If you select Deny, the router denies the route from entering the BGP4 table if the filter match is true.
   - If you select Permit, the router permits the route into the BGP4 table if the filter match is true.
7. Enter the network prefix in the Prefix field. If you specify “any”, all networks match the filter.
8. Enter the prefix masking bits in the Prefix Masking Bits field. The prefix masking bits indicate the bits in the prefix that the filter compares. The filter disregards the bits for which the mask contains zeros.
9. Enter the mask in the Prefix Mask field. If you specify “any”, all masks match the filter.
10. Enter the masking bits for the network mask in the Prefix Mask Masking Bits field.
11. Click the Add button to apply the changes to the device’s running-config file.
12. Repeat steps 5 – 11 for each address filter.
13. In the tree view, under BGP in the Configure section, click on the Route Map Filter link to display the BGP Route Map Filter panel.
    - If the device does not have any BGP route map filters configured, the BGP Route Map Filter configuration panel is displayed, as shown in the following example.
    - If BGP route map filters are already configured and you are adding a new one, click on the Route Map Filter link to display the BGP Route Map Filter configuration panel, as shown in the following example.
    - If you are modifying an existing BGP route map filter, click on the Modify button to the right of the row describing the filter to display the BGP Route Map Filter configuration panel, as shown in the following

example.

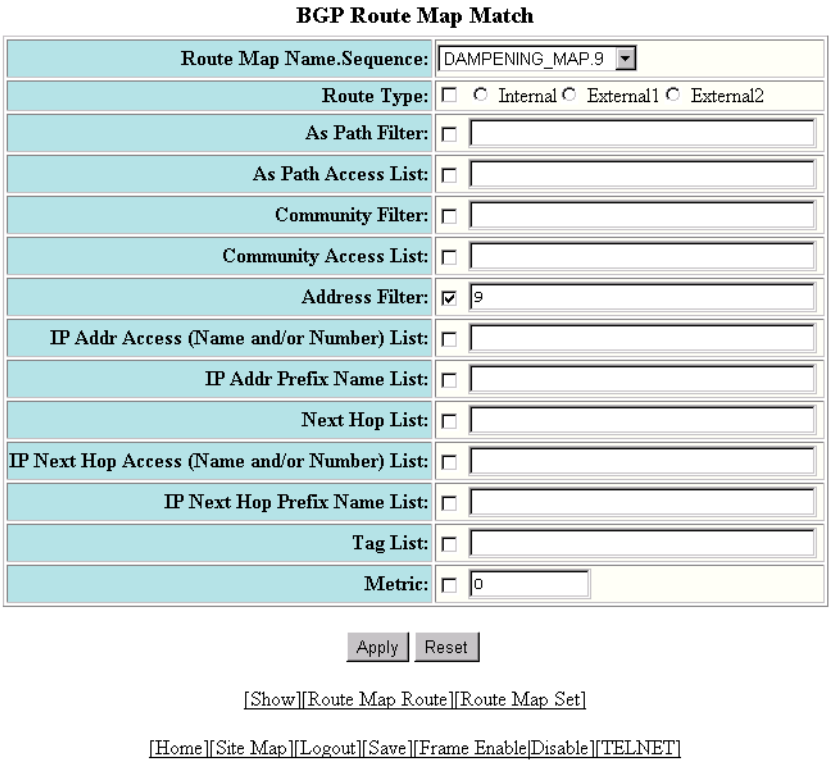


14. Enter the name of the route map in the Route Map Name field.
15. Enter the sequence (instance) number in the Sequence field. The Layer 3 Switch applies the instances in ascending numerical order. Once an instance comparison results in a “true” evaluation, the Layer 3 Switch stops applying instances and applies the match and set statements you configure for the instance.

    NOTE: In this example, the sequence number matches the address filter number. Using the same number is a convenient way to remember that these configuration items are associated, but is not a requirement.

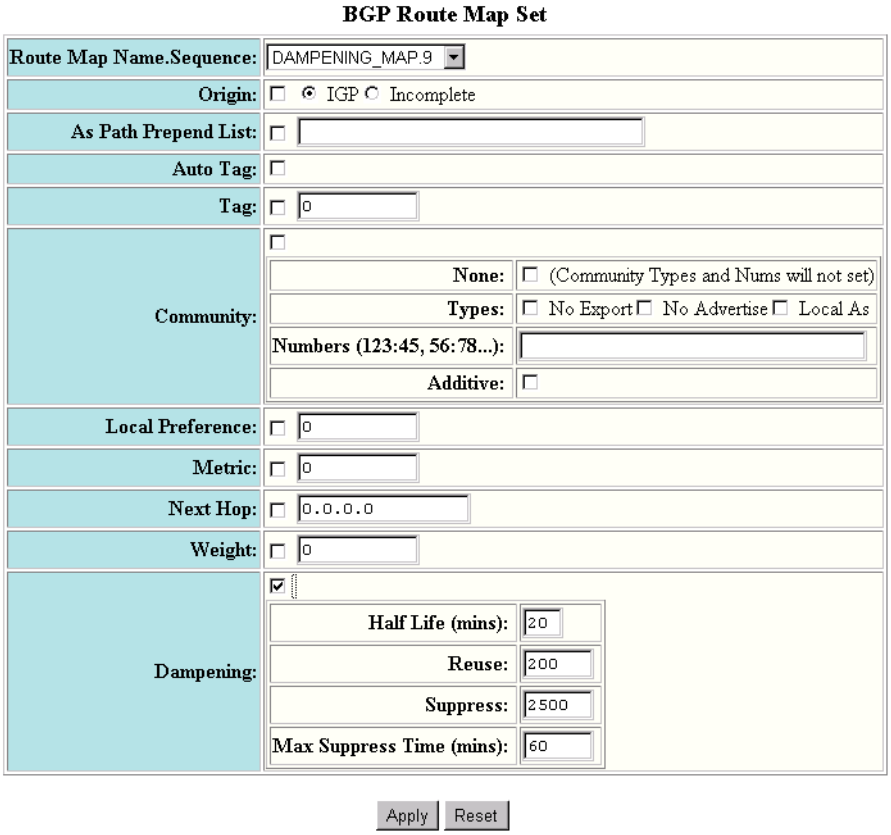
16. Select the action you want the Layer 3 Switch to perform if the comparison results in a “true” value:
    - If you select Deny, the Layer 3 Switch does not advertise or learn the route.
    - If you select Permit, the Layer 3 Switch applies the match and set statements associated with this route map instance.
17. Click the Add button to apply the changes to the device’s running-config file.

18. Select the Route Map Match link at the bottom of the panel to display the BGP Route Map Match panel, as shown in the following example.



19. Click on the checkbox next to Address Filter to indicate that you are using an address filter as a match condition.
20. Enter the address filter number in the Address Filter field.
21. Click Apply to apply the changes to the device's running-config file.

22. Select the Route Map Set link at the bottom of the panel to display the BGP Route Map Set panel, as shown in the following example.



23. Select the checkbox in the Dampening section to specify that this route map is setting dampening parameters.
24. Edit the value in the Half Life field to specify the half life you want this route map to set for routes that match the match conditions you specified above.
25. Edit the value in the Reuse field to specify the dampening reuse value you want this route map to set.
26. Edit the value in the Suppress field to specify the dampening suppress value you want this route map to set.
27. Edit the value in the Max Suppress Time field to specify the maximum suppression value you want this route map to set.
28. Click Apply to apply the changes to the device's running-config file.
29. In the tree view, under BGP in the Configure section, click on the General link to display the BGP configuration panel.
30. In the Dampening section, click next to Route-Map, then select the dampening route map from the Route-Map field's pulldown menu. In this example, select the map named DAMPENING_MAP.

NOTE: The route map appears in this menu only if you have already configured the route map.

31. Click Apply to apply the changes to the device's running-config file.
32. 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.

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.

USING THE CLI

To enable route flap dampening for a specific BGP4 neighbor, enter commands such as the following:

```
BigIron(config)# route-map DAMPENING_MAP_ENABLE permit 1
BigIron(config-routemap DAMPENING_MAP_ENABLE)# exit
BigIron(config)# route-map DAMPENING_MAP_NEIGHBOR_A permit 1
BigIron(config-routemap DAMPENING_MAP_NEIGHBOR_A)# set dampening
BigIron(config-routemap DAMPENING_MAP_NEIGHBOR_A)# exit
BigIron(config)# router bgp
BigIron(config-bgp-router)# dampening route-map DAMPENING_MAP_ENABLE
BigIron(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.

USING THE WEB MANAGEMENT INTERFACE

1. Log on to the device using a valid user name and password for read-write access. The System configuration panel is displayed.
2. Click on the plus sign next to Configure in the tree view to expand the list of configuration options.
3. Click on the plus sign next to BGP in the tree view to expand the list of BGP option links.

4. In the tree view, under BGP in the Configure section, click on the Route Map Filter link to display the BGP Route Map Filter panel.

   NOTE: If the device already has route maps, a table listing the route maps is displayed. Click the Modify button to the right of the row describing the route map to change its configuration, or click the Add Route Map Filter link to display the BGP Route Map Filter panel.

5. Enter the name of the route map in the Route Map Name field. In this example, enter the name DAMPENING_MAP_ENABLE for the "empty" route map that you will use to globally enable dampening.
6. Enter the sequence (instance) number in the Sequence field or use the default value.
7. Select the action you want the Layer 3 Switch to perform if the comparison results in a "true" value:
   - If you select Deny, the Layer 3 Switch does not advertise or learn the route.
   - If you select Permit, the Layer 3 Switch applies the match and set statements associated with this route map instance. In this example, select Permit.
8. Click the Add button to apply the changes to the device's running-config file.

   NOTE: In this case, you are configuring an "empty" route map with no match or set statements, so you do not need to select the Route Map Match or Route Map Set link.

9. Enter the name of the route map you will use to set dampening parameters for a neighbor in the Route Map Name field. In this example, enter the name DAMPENING_MAP_NEIGHBOR_A.
10. Select the action you want the Layer 3 Switch to perform if the comparison results in a "true" value:
    - If you select Deny, the Layer 3 Switch does not advertise or learn the route.
    - If you select Permit, the Layer 3 Switch applies the match and set statements associated with this route map instance. In this example, select Permit.
11. Click the Add button to apply the changes to the device's running-config file.
12. Select the Route Map Set link to display the BGP Route Map Set panel.

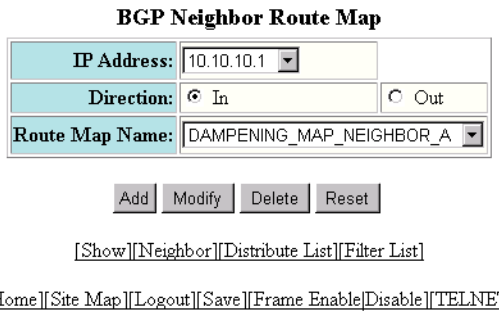
    NOTE: If the interface displays a table listing the configured route maps, select the Route Map Set link under the table or click Modify next to the row describing the route map you are configuring.

13. Select the route map name and sequence from the Route Map Name.Sequence field's pulldown menu.
14. Select the checkbox in the Dampening section to enable dampening for routes that match the route map.
15. Click the Apply button to apply the changes to the device's running-config file.
16. In the tree view, under BGP in the Configure section, click on the General link to display the BGP configuration panel.
17. In the Dampening section, click next to Route-Map, then select the dampening route map from the Route-Map field's pulldown menu. In this example, select the map named DAMPENING_MAP_ENABLE.

    NOTE: The route map appears in this menu only if you have already configured the route map.

18. Click Apply to apply the changes to the device's running-config file.
19. In the tree view, under BGP in the Configure section, click on the Neighbor link to display the list of BGP neighbors.
20. Select the Modify button to the right of the row describing the neighbor to which you want to apply the dampening route map you configured in steps 9 – 15.

21. Select the Route Map link at the bottom of the panel to display the BGP Neighbor Route Map panel, as shown in the following example.



22. Select the neighbor IP address from the IP Address field's pulldown menu.
23. Select the traffic direction to which you want to apply the route map. You can select In or Out. In this example, select In.
24. Select the route map from the Route Map Name field's pulldown menu. In this example, select DAMPENING_MAP_NEIGHBOR_A.
25. Click Add to apply the changes to the device's running-config file.
26. 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.

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 routes, use either of the following methods.

USING THE CLI

To un-suppress all the suppressed routes, enter the following command at the Privileged EXEC level of the CLI:

```
BigIron# 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:

```
BigIron# 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.

USING THE WEB MANAGEMENT INTERFACE

1. Log on to the device using a valid user name and password for read-write access. The System configuration panel is displayed.
2. Click on the plus sign next to Command in the tree view to expand the list of command options.
3. Click on the Clear link to display the Clear panel.
4. Select the checkbox next to BGP Dampening.

5. Specify the routes from which you want to remove dampening:
   - To clear dampening for all routes, select the All option.
   - To clear dampening for a specific route, select IP, then enter the network address and subnet mask in the IP and Mask fields.
6. Click the Apply button to implement the change.

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.

```
BigIron(config-bgp-router)# aggregate-address 209.1.0.0 255.255.0.0 summary-only
BigIron(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.

```
BigIron(config-bgp-router)# 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:

```
BigIron(config)# ip prefix-list Unsuppress1 permit 209.1.44.0/24
BigIron(config)# route-map RouteMap1 permit 1
BigIron(config-routemap RouteMap1)# match prefix-list Unsuppress1
BigIron(config-routemap RouteMap1)# exit
BigIron(config)# router bgp
BigIron(config-bgp-router)# neighbor 10.1.0.2 unsuppress-map RouteMap1
BigIron(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.

```
BigIron(config-bgp-router)# 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 flap dampening statistics, use the following CLI method.

USING THE CLI

To display route dampening statistics or all the dampened routes, enter the following command at any level of the CLI:

```
BigIron# 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 16-63.

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 16.3: 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.
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.

USING THE WEB MANAGEMENT INTERFACE

You cannot display dampening statistics using the Web management interface.

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.

USING THE CLI

To clear all the route dampening statistics, enter the following command at any level of the CLI:

```
BigIron# 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 16-106.

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 16-106.

USING THE WEB MANAGEMENT INTERFACE

You cannot clear dampening statistics using the Web management interface.

Statically Allocating Memory in Earlier Software Releases

NOTE: These procedures apply only to the TurboIron/8 or NetIron stackable running a software release earlier than 07.1.00 or to the BigIron or NetIron Internet Backbone router running a release earlier than 07.0.00. These releases use static memory allocation for BGP4. For later software releases, these procedures are unnecessary and are not supported. See "Memory Considerations" on page 16-10.

Changing the Maximum Number of Neighbors

You can change the maximum number of BGP4 neighbors the Layer 3 Switch can have using either of the following methods.

NOTE: If you have a lot of IBGP neighbors, you can configure some IBGP routers as route reflectors. By doing so, you can reduce the number of neighbors you need to configure on each router. Without route reflectors, all IBGP routers must be fully meshed to ensure proper route propagation. See "Configuring Route Reflection Parameters" on page 16-42.

USING THE CLI

To change the maximum number of BGP4 neighbors to 3, enter the following command:

```
NetIron(config-bgp-router)# max-neighbors 3
NetIron(config-bgp-router)# end
NetIron# reload
```

Syntax: max-neighbors <num>

The <num> indicates the number of BGP4 neighbors allowed. See "Memory Considerations" on page 16-10 for the maximum for your device. The change takes effect after the router is rebooted.

USING THE WEB MANAGEMENT INTERFACE

1. Log on to the device using a valid user name and password for read-write access. The System configuration panel is displayed.
2. Click on the plus sign next to Configure in the tree view to expand the list of configuration options.
3. Click on the plus sign next to BGP in the tree view to expand the list of BGP option links.
4. Click on the General link to display the BGP configuration panel, shown in Figure 16.2 on page 16-9.
5. Change the number in the Maximum Neighbors field. The maximum number you can enter depends on the device you are configuring. See "Memory Considerations" on page 16-10 for the maximum for your device.
6. Click the Apply button to apply the changes to the device's running-config file.
7. 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.
8. Click on Command in the tree view to list the command options.
9. Select the Reload link and click on Yes when prompted. You must reload the software to place this configuration change into effect.

Changing the Maximum Number of Routes

You can change the maximum number of BGP4 routes the router can have using either of the following methods.

NOTE: This value also determines the maximum value you can configure when specifying how many routes this Layer 3 Switch can learn from all its neighbors. See the description of the maximum prefix option in “Adding BGP4 Neighbors” on page 16-14.

USING THE CLI

To change the maximum number of BGP4 routes to 30000, enter the following command:

```
NetIron(config-bgp-router)# max-routes 30000
NetIron(config-bgp-router)# end
NetIron# reload
```

Syntax: max-routes <num>

The <num> indicates the number of BGP4 routes allowed. See “Memory Considerations” on page 16-10 for the maximum for your device. The change takes effect after the router is rebooted.

USING THE WEB MANAGEMENT INTERFACE

1. Log on to the device using a valid user name and password for read-write access. The System configuration panel is displayed.
2. Click on the plus sign next to Configure in the tree view to expand the list of configuration options.
3. Click on the plus sign next to BGP in the tree view to expand the list of BGP option links.
4. Click on the General link to display the BGP configuration panel, shown in Figure 16.2 on page 16-9.
5. Change the number in the Maximum Routes field. The maximum number you can enter depends on the device you are configuring. See “Memory Considerations” on page 16-10 for the maximum for your device.
6. Click the Apply button to apply the changes to the device’s running-config file.
7. 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.
8. Click on Command in the tree view to list the command options.
9. Select the Reload link and click on Yes when prompted. You must reload the software to place this configuration change into effect.

Changing the Maximum Number of Route-Attribute Entries

The BGP4 route table lists the route attributes associated with each route in the table. These attributes include the following:

- IP address of the next hop router
- Metric
- Local Preference
- Origin
- Communities
- and others

A collection of these attributes is called a ***route-attributes entry***. Each route-attributes entry is a unique set of values for these attributes. For example, the following set of attribute values is a route-attributes entry:

```
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
```

A route-attribute entry can be used by one or more routes. For example, if the first and second routes listed in the BGP4 route table use exactly the same set of attribute values, the routes both would use a single route-attributes entry. If any of the attributes differs for the two routes, each route would use a separate route-attributes entry. See "Displaying BGP4 Route-Attribute Entries" on page 16-142 for a description of the route-attribute fields shown in the example above.

You can change the maximum number of route-attribute entries the router can contain using either of the following methods.

USING THE CLI

To change the maximum number of route-attribute entries to 2500, enter the following command:

```
NetIron(config-bgp-router)# max-attribute-entries 2500
NetIron(config-bgp-router)# end
NetIron# reload
```

Syntax: max-attribute-entries <num>

The <num> indicates the number of route-attribute entries allowed on the router. See "Memory Considerations" on page 16-10 for the maximum for your device. The change takes effect after the router is rebooted.

USING THE WEB MANAGEMENT INTERFACE

1. Log on to the device using a valid user name and password for read-write access. The System configuration panel is displayed.
2. Click on the plus sign next to Configure in the tree view to expand the list of configuration options.
3. Click on the plus sign next to BGP in the tree view to expand the list of BGP option links.
4. Click on the General link to display the BGP configuration panel, shown in Figure 16.2 on page 16-9.
5. Change the number in the Maximum Attribute Entries field. The maximum number you can enter depends on the device you are configuring. See "Memory Considerations" on page 16-10 for the maximum for your device.
6. Click the Apply button to apply the changes to the device's running-config file.
7. 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.
8. Click on Command in the tree view to list the command options.
9. Select the Reload link and click on Yes when prompted. You must reload the software to place this configuration change into effect.

Generating Traps for BGP

Software release 07.7.00 provides the ability to 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:

```
BigIron(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 using either of the following methods.

USING THE CLI

To view summary BGP4 information for the router, enter the following command at any CLI prompt:

```
BigIron# 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 16.4: BGP4 Summary Information

This Field...	Displays...
Router ID	The Layer 3 Switch's router ID.
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 – 8 paths. See "Changing the Maximum Number of Paths for BGP4 Load Sharing" on page 16-29.
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 16-134.
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 16-142.
Neighbor Address	The IP addresses of this router's BGP4 neighbors.
AS#	The AS number.

Table 16.4: 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 16-26.
	  • 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 16.4: 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.

USING THE WEB MANAGEMENT INTERFACE

1. Log on to the device using a valid user name and password for read-only or read-write access. The System configuration panel is displayed.
2. Click on the plus sign next to Monitor in the tree view to expand the list of monitoring options.
3. Click on the plus sign next to BGP in the tree view to expand the list of BGP option links.
4. Click on the Summary link to display the BGP Neighbor Summary panel.

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.

USING THE CLI

To display the device's active BGP4 configuration, enter the following command at any level of the CLI:

```
BigIron# 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 "BigIron 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

USING THE WEB MANAGEMENT INTERFACE

You cannot display the BGP4 running-config information using the Web management interface.

Displaying CPU Utilization Statistics

You can display CPU utilization statistics for BGP4 and other IP protocols.

USING THE CLI

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:

```
BigIron# 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:

```
BigIron# 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:

```
BigIron# 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.

USING THE WEB MANAGEMENT INTERFACE

You cannot display this information using the Web management interface.

Displaying Summary Neighbor Information

To display information for a neighbor, use the following CLI method.

USING THE CLI

To display summary neighbor information, enter a command such as the following at any level of the CLI:

```
BigIron(config-bgp-router)# 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 16.5: BGP4 Route Summary Information for a Neighbor

This Field...	Displays...
IP Address	The IP address of the neighbor

Table 16.5: BGP4 Route Summary Information for a Neighbor (Continued)

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/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.
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.

Table 16.5: BGP4 Route Summary Information for a Neighbor (Continued)

This Field...	Displays...
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.
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.

USING THE WEB MANAGEMENT INTERFACE

You cannot display summary neighbor information using the Web management interface.

Displaying BGP4 Neighbor Information

You can display configuration information and statistics for the router’s BGP4 neighbors using either of the following methods.

USING THE CLI

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.

```
BigIron(config-bgp-router)# 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 16-147.

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 16.6: BGP4 Neighbor Information

This Field...	Displays...
IP Address	The IP address of the neighbor.
AS	The AS the neighbor is in.

Table 16.6: 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 16.6: 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 16-26.
	  • 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 16-27.
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 16-27.
PeerGroup	The name of the peer group the neighbor is in, if applicable.

Table 16.6: 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 16.6: 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 16.6: 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 16.6: 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 16.6: 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.
TotSent	The number of sequence numbers sent to the neighbor.

Table 16.6: BGP4 Neighbor Information (Continued)

This Field...	Displays...
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.

USING THE CLI

Displaying Summary Route Information

To display summary route information, enter a command such as the following at any level of the CLI:

```
BigIron(config-bgp-router)# 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 16.7: 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 16.7: 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 16.7: 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:

```
BigIron# 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:

```
BigIron# 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 16.9 on page 16-138. 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:

```
BigIron(config-bgp-router)# 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 16.9 on page 16-138. 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:

```
BigIron(config-bgp-router)# 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 16.9 on page 16-138. 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:

```
BigIron(config-bgp-router)# 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 16.9 on page 16-138. 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:

```
BigIron(config-bgp-router)# 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 16.9 on page 16-138. The fields in this display also appear in the **show ip bgp** display.

USING THE WEB MANAGEMENT INTERFACE

1. Log on to the device using a valid user name and password for read-only or read-write access. The System configuration panel is displayed.
2. Click on the plus sign next to Monitor in the tree view to expand the list of monitoring options.
3. Click on the plus sign next to BGP in the tree view to expand the list of BGP option links.
4. Click on the Neighbor link to display the BGP Neighbor Statistics panel.

Displaying Peer Group Information

You can display configuration information for peer groups.

USING THE CLI

To display peer-group information, enter a command such as the following at the Privileged EXEC level of the CLI:

```
BigIron# 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 route information, use the following CLI method.

USING THE CLI

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:

```
BigIron(config-bgp-router)# 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 16.8: 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 16-147.
Routes originated by this router	The number of routes in the BGP4 route table that this Layer 3 Switch originated.

Table 16.8: BGP4 Summary Route Information (Continued)

This Field...	Displays...
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.

USING THE WEB MANAGEMENT INTERFACE

You cannot display summary route information using the Web management interface.

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 16-3 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.

USING THE CLI

To view the BGP4 route table, enter the following command:

```
BigIron(config-bgp-router)# 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 16-63.

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:

```
BigIron(config-bgp-router)# 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 16.9 on page 16-138. 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:

```
BigIron(config-bgp-router)# 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 16.9 on page 16-138 for a description.

Syntax: show ip bgp routes not-installed-best

For information about the fields in this display, see Table 16.9 on page 16-138. 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:

```
BigIron(config-bgp-router)# 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 16.9 on page 16-138. The fields in this display also appear in the **show ip bgp** display.

Displaying Information for a Specific Route

To display information for a specific BGP4 route, use either of the following methods.

USING THE CLI

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:

```
BigIron(config-bgp-router)# 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:

```
BigIron(config-bgp-router)# 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 16.9: 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.
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 16.9: 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.

```
BigIron# 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 16.10: 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 16.10: 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.

USING THE WEB MANAGEMENT INTERFACE

1. Log on to the device using a valid user name and password for read-only or read-write access. The System configuration panel is displayed.
2. Click on the plus sign next to Monitor in the tree view to expand the list of monitoring options.
3. Click on the plus sign next to BGP in the tree view to expand the list of BGP option links.
4. Click on the Routes link to display the BGP Routes panel.

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.

USING THE CLI

To display the IP route table, enter the following command:

```
BigIron# 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.

```
BigIron# 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 16.11: 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.

Table 16.11: BGP4 Route-Attribute Entries Information (Continued)

This Field...	Displays...
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.
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.

USING THE WEB MANAGEMENT INTERFACE

1. Log on to the device using a valid user name and password for read-only or read-write access. The System configuration panel is displayed.
2. Click on the plus sign next to Monitor in the tree view to expand the list of monitoring options.
3. Click on the plus sign next to BGP in the tree view to expand the list of BGP option links.
4. Click on the Attributes link to display the BGP Attributes Entries panel.

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. You can view the IP route table using either of the following methods.

USING THE CLI

To display the IP route table, enter the following command:

```
BigIron# 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.

```
BigIron# 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...
```

USING THE WEB MANAGEMENT INTERFACE

1. Log on to the device using a valid user name and password for read-only or read-write access. The System configuration panel is displayed.
2. Click on the plus sign next to Monitor in the tree view to expand the list of monitoring options.
3. Click on the plus sign next to IP in the tree view to expand the list of IP option links.
4. Click on the Routing Table link to display the IP route table.

Displaying Route Flap Dampening Statistics

To display route flap dampening statistics, use the following CLI method.

USING THE CLI

To display route dampening statistics or all the dampened routes, enter the following command at any level of the CLI:

```
BigIron# 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 16-63.

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 16.12: 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.
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**.

USING THE WEB MANAGEMENT INTERFACE

You cannot display dampening statistics using the Web management interface.

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, use the following CLI method.

USING THE CLI

To display the device's active route map configuration, enter the following command at any level of the CLI:

```
BigIron# 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:

```
BigIron# 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>]

USING THE WEB MANAGEMENT INTERFACE

You cannot display the active route map configuration using the Web management interface.

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 reestablish 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 16-153.

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.

USING THE CLI

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:

```
BigIron(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 16-14.

Placing a Policy Change into Effect

To place policy changes into effect, enter a command such as the following:

```
BigIron(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 16-150.

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:

```
BigIron# 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 16-134.

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:

```
BigIron# 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 16-118.

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 when you upgrade to software release 07.1.00 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.

USING THE CLI

To request a dynamic refresh of all routes from a neighbor, enter a command such as the following:

```
BigIron(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 16-147.
  - 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:

```
BigIron(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**).

USING THE WEB MANAGEMENT INTERFACE

You cannot perform these reset procedures using the Web management interface.

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.

```
BigIron(config-bgp-router)# 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.

USING THE CLI

To close a neighbor session and thus flush all the routes exchanged by the Layer 3 Switch and the neighbor, enter the following command:

```
BigIron# 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:

```
BigIron# clear ip bgp neighbor 10.0.0.1 soft out
```

USING THE WEB MANAGEMENT INTERFACE

To resend route information to a neighbor, use the following procedure:

1. Log on to the device using a valid user name and password for read-write access. The System configuration panel is displayed.
2. Click on the plus sign next to Command in the tree view to expand the list of command options.
3. Click on the Clear link to display the Clear panel.
4. Select BGP Neighbor Soft-Outbound.
5. Use the default value All to resend the BGP4 route table to all neighbors or select a neighbor from the field's pulldown menu.
6. Click the Apply button to implement the change.

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:

```
BigIron# 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.

USING THE CLI

To clear the BGP4 message counter for all neighbors, enter the following command:

```
BigIron# 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:

```
BigIron# 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:

```
BigIron# 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.

USING THE WEB MANAGEMENT INTERFACE

1. Log on to the device using a valid user name and password for read-write access. The System configuration panel is displayed.
2. Click on the plus sign next to Command in the tree view to expand the list of command options.
3. Click on the Clear link to display the Clear panel.
4. Select one of the following options:
   - BGP Neighbor Traffic – clears the BGP4 message counters for all neighbors (the default) or a neighbor you select from the pulldown menu.
   - BGP Neighbor – clears the BGP4 message counters for all neighbors (the default) or a neighbor you select from the pulldown menu.
5. Click the Apply button to implement the change.

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.

USING THE CLI

To clear all the route dampening statistics, enter the following command at any level of the CLI:

```
BigIron# 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 16-106.

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 16-106.

USING THE WEB MANAGEMENT INTERFACE

You cannot clear dampening statistics using the Web management interface.

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 routes, use either of the following methods.

USING THE CLI

To un-suppress all the suppressed routes, enter the following command at the Privileged EXEC level of the CLI:

```
BigIron# 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:

```
BigIron# 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.

USING THE WEB MANAGEMENT INTERFACE

1. Log on to the device using a valid user name and password for read-write access. The System configuration panel is displayed.
2. Click on the plus sign next to Command in the tree view to expand the list of command options.
3. Click on the Clear link to display the Clear panel.
4. Select the checkbox next to BGP Dampening.
5. Specify the routes from which you want to remove dampening:
   - To clear dampening for all routes, select the All option.
   - To clear dampening for a specific route, select IP, then enter the network address and subnet mask in the IP and Mask fields.
6. Click the Apply button to implement the change.

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 16-118.

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.

USING THE CLI

To clear these buffers for neighbor 10.0.0.1, enter the following commands:

```
BigIron# clear ip bgp neighbor 10.0.0.1 last-packet-with-error
BigIron# 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.

USING THE WEB MANAGEMENT INTERFACE

1. Log on to the device using a valid user name and password for read-write access. The System configuration panel is displayed.
2. Click on the plus sign next to Command in the tree view to expand the list of command options.
3. Click on the Clear link to display the Clear panel.
4. Select one of the following:
   - BGP Neighbor Last Packet with Error – Clears the buffer containing the first 400 bytes of the last BGP4 packet that contained an error.
   - BGP Neighbor Notification Error – Clears the buffer containing the last NOTIFICATION message sent or received.
5. Click the Apply button to implement the change.

Chapter 17
Configuring MBGP

This chapter provides details on how to configure ***Multi-protocol Border Gateway Protocol (MBGP)***. MBGP is an extension to BGP that allows a router to support separate unicast and multicast topologies. BGP4 cannot support a multicast network topology that differs from the network's unicast topology. MBGP allows you to support a multicast topology that is distinct from the network's unicast topology. For example, if you want to dedicate a link on your Internet router to multicast traffic, use MBGP to handle the routes on that link.

MBGP provides the following benefits:

- You can support a network whose multicast topology is different from its unicast topology. Even if the unicast and multicast networks have the same topologies, you can support different sets of routing policies for unicast and multicast.
- You can use BGP4's powerful feature set with MBGP.

MBGP is supported on the following Foundry products:

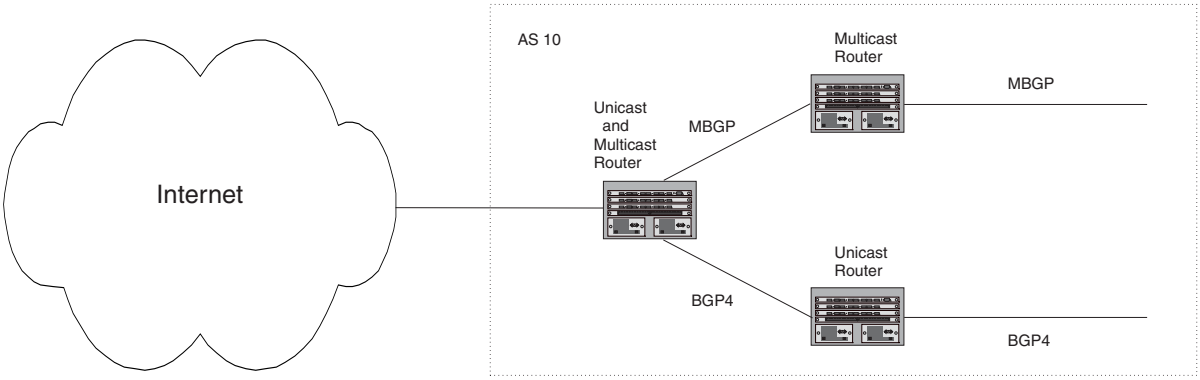
- NetIron Internet Backbone router
- BigIron Layer 3 Switch
- BigIron MG8 and NetIron 40G running software release 02.2.01.

NOTE: MBGP is not supported on any FastIron products.

Overview

Figure 17.1 shows an example of a network that contains both a unicast topology and a multicast topology. The unicast and multicast router in this example receives unicast and multicast routes from the Internet. The router advertises the multicast routes to the multicast router and advertises the unicast routes to the unicast router. Likewise, the unicast and multicast router can advertise unicast routes received from the unicast router to the Internet, and can advertise multicast routes received from the multicast router to the Internet.

Figure 17.1 MBGP used when multicast topology is different from unicast topology



An MBGP router learns MBGP routes from its neighbors in other ASs. An MBGP router also can advertise MBGP routes to its neighbors. The Foundry implementation of MBGP enables you to advertise multicast routes from the following sources:

- Explicitly configured network prefixes
- Static IP multicast routes
- Directly-connected multicast routes redistributed into MBGP using a route map

You can configure an aggregate address to aggregate network prefixes into a single, more general prefix for advertisement.

NOTE: Foundry's implementation of MBGP supports redistributing routes into Protocol Independent Multicast (PIM), but not into Distance Vector Multicast Routing Protocol (DVMRP).

MBGP is described in detail in RFC 2858.

Configuration Considerations

- MBGP does not redistribute DVMRP routes. Only PIM SM and PIM DM routes are redistributed.
- You cannot redistribute MBGP routes into BGP4.
- Confederations, route reflection, route flap dampening, and outbound route filters are not supported in this release.
- The Layer 3 Switch supports up to 16 multicast routes by default. You may need to increase the maximum number of multicast routes for MBGP. You can configure the device to support up to 8192 multicast routes.

Configuring MBGP

1. Optional – Set the maximum number of multicast routes supported by the Layer 3 Switch.

   NOTE: If MBGP is enabled, set this number to the number of multicast routes being received.

2. Enable MBGP by doing the following:
   - Enable PIM Sparse Mode (PIM SM) or PIM Dense Mode (PIM DM) globally and on the individual Reverse Path Forwarding (RPF) interfaces. PIM must be running on the Layer 3 Switch in order for the device to send multicast prefixes to other multicast routers.
   - Enable BGP4. If this is the first time you have configured BGP4 on this device, you also need to specify the local AS number.
3. Identify the neighboring MBGP routers.

4. Optional – Configure an MBGP default route.
5. Optional – Configure an IP multicast static route.
6. Optional – Configure an MBGP aggregate address.
7. Optional – Configure a route map to apply routing policy to multicast routes.
8. Save the configuration changes to the startup-config file.

Setting the Maximum Number of Multicast Routes Supported

The Layer 3 Switch supports up 16 – 8,000 multicast routes. If MBGP is enabled, set this number to the number of multicast routes being received.

NOTE: This procedure requires a software reload to place the change into effect.

To increase the maximum number of multicast routes supported on the device, enter commands such as the following:

```
BigIron(config)# system-max multicast-route 2048
BigIron(config)# write memory
BigIron(config)# end
BigIron# reload
```

These commands increase the maximum number of multicast routes supported, save the configuration change to the startup-config file, and reload the software to place the change into effect.

Syntax: [no] system-max multicast-route <num>

The <num> parameter specifies the number of multicast routes and can be from 16 – 8192.

Enabling MBGP

To enable MBGP4, you must enable PIM SM or DM and BGP4. Enter commands such as the following:

```
BigIron> enable
BigIron# configure terminal
BigIron(config)# router pim
PIM enabled for next power cycle.
Please save configuration to flash and reboot.
BigIron(config-pim-router)# write memory
BigIron(config-pim-router)#.Write startup-config in progress.
.Write startup-config done.
BigIron(config-pim-router)# end
BigIron# reload
Are you sure? (enter 'y' or 'n'): y
Halt and reboot
BOOT INFO: RESET ACTIVE
<remaining boot messages omitted for brevity>

BigIron> enable
BigIron# configure terminal
BigIron(config)# interface ethernet 1/1
BigIron(config-if-1/1)# ip address 1.1.1.1/24
BigIron(config-if-1/1)# ip pim
BigIron(config-if-1/1)# exit
BigIron(config)# router bgp
BGP4: Please configure 'local-as' parameter in order to enable BGP4.
BigIron(config-bgp-router)# local-as 10
BigIron(config-bgp-router)# write memory
```


The commands in this example configure PIM DM globally and on port 1/1, then enable BGP4. Once you enable PIM DM or PIM SM both globally and on the individual RPF interfaces, and enable BGP4, support for MBGP is automatically enabled.

Adding MBGP Neighbors

To add an MBGP neighbor, enter a command such as the following:

```
BigIron(config-bgp-router)# neighbor 1.2.3.4 remote-as 44 nlri multicast
```

This command adds a router with IP address 1.2.3.4 as an MBGP neighbor.

The **remote-as 44** command specifies that the neighbor is in remote BGP4 AS 44. The **nlri multicast** parameter specifies that the router is an MBGP neighbor only (multicast), not a BGP4 neighbor (unicast). The Layer 3 Switch will exchange only multicast routes with the neighbor.

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.

Here is the full syntax for the neighbor command. The parameters required for MBGP configuration are explained following the syntax.

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]
[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>]
[update-source loopback <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.

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 **remote-as** <as-number> parameter specifies the AS the MBGP neighbor is in. The <as-number> can be a number from 1 – 65535. There is no default.

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.

Optional Configuration Tasks

The following sections describe how to perform some optional BGP4 configuration tasks.

NOTE: This section shows some of the more common optional tasks, including all the tasks that require you to specify that they are for MBGP (by setting the NLRI type to multicast). Most tasks are configured only for BGP4 but apply both to BGP4 and MBGP. For information on these other tasks, see "Configuring BGP4" on page 16-1.

NOTE: For tasks that allow you to specify the NLRI type, the default is unicast. For MBGP, make sure you specify the NLRI type as multicast.

Advertising Routes from the Local AS to MBGP

You can configure the Layer 3 Switch to advertise directly-connected and indirectly-connected multicast routes from the local AS to other ASs using the following methods:

- For directly-connected routes:
  - Enable redistribution of directly-connected multicast routes using route maps. Set the NLRI to multicast.
- For indirectly-connected routes:
  - Configure static IP multicast routes. The corresponding IP route must be present in the IP route table.
  - Explicitly configure network prefixes to advertise (**network** command).

NOTE: You can configure the device to advertise directly-connected networks into MBGP using the **network** command. You are not required to use redistribution or configure static multicast routes.

Configuring a Network Prefix to Advertise

By default, the Layer 3 Switch advertises MBGP routes only for the networks you identify using the **network** command or that are redistributed into MBGP from IP multicast static routes. You can specify up to 600 network prefixes for advertisement.

NOTE: The exact route must exist in the IP multicast route table and IP route table before the Layer 3 Switch can create a local MBGP route.

To configure the Layer 3 Switch to advertise network 207.95.22.0/24 as a multicast route, enter the following command:

```
BigIron(config-bgp-router)# network 207.95.22.0 255.255.255.0 nlri multicast
```

Syntax: network <ip-addr> <ip-mask> [nlri multicast | unicast | multicast unicast]
[route-map <map-name>]

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.

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.

Enabling Redistribution of Directly-Connected Multicast Routes into MBGP

To redistribute a directly-connected multicast route into MBGP, configure a route map, then enable redistribution of directly-connected routes into MBGP, using the route map to specify the routes to be redistributed. Here is an example.

```
BigIron(config)# access-list 10 permit 207.95.22.0 0.0.0.255
BigIron(config)# route-map mbgpmap permit 1
BigIron(config-routemap mbgpmap)# match nlri multicast
BigIron(config-routemap mbgpmap)# match ip address 10
BigIron(config-routemap mbgpmap)# set nlri multicast
BigIron(config-routemap mbgpmap)# exit
BigIron(config)# router bgp
BigIron(config-bgp-router)# redistribute connected route-map mbgpmap
```

The first command configures an IP ACL for use in the route map. The ACL matches on the destination network for the route to be redistributed. The next four commands configure a route map that matches on routes to the multicast network specified in IP ACL 10. The **match nlri multicast** command configures the route map to match only on NLRI type multicast. By default, route maps match on both unicast and multicast. The **match ip address 10** command configures the route map to match on the IP address specified in IP ACL 10. The **set nlri multicast** command configures the route map to set the NLRI type of the route to multicast. The Layer 3 Switch redistributes routes that match the route map into MBGP as multicast routes.

Syntax: [no] redistribute connected [metric <num>] [route-map <map-name>] [weight <num>]

The **connected** parameter indicates that you are redistributing routes to directly attached devices into MBGP.

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 filter to the IP route table.

NOTE: The route map you specify must already be configured.

The **weight** <num> parameter changes the weight. You can specify a value from 0 – 65535. The default is 0.

Configuring Static IP Multicast Routes for Redistribution into MBGP

The Foundry implementation of MBGP can redistribute static IP multicast routes into MBGP.

NOTE: Redistribution into MBGP also requires a static unicast route for the static multicast route.

To configure static IP multicast routes, enter commands such as the following:

```
BigIron(config)# ip mroute 1 207.95.10.0 255.255.255.0 interface ethernet 1/2
distance 1
BigIron(config)# ip mroute 2 0.0.0.0 0.0.0.0 interface ethernet 2/3 distance 1
```

The commands in this example configure 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.

If you 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 this example.

Syntax: mroute <route-num> <ip-addr> interface ethernet <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.

You can use the **ethernet** <portnum> parameter to specify a physical port or the **ve** <num> parameter to specify a virtual interface.

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.

Aggregating Routes Advertised to BGP4 Neighbors

By default, the Layer 3 Switch advertises individual MBGP routes for all the multicast 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 207.95.10.0, 207.95.20.0, and 207.95.30.0. 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.

To aggregate MBGP routes for 207.95.10.0, 207.95.20.0, and 207.95.30.0, enter the following command:

```
BigIron(config-bgp-router)# aggregate-address 207.95.0.0 255.255.0.0
```

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.

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 Layer 3 Switch to advertise the more specific routes in the specified route map.

The **attribute-map** <map-name> parameter configures the Layer 3 Switch 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.

Using Route Maps

You can use route maps to filter and change values in MBGP routes. ***By default, route maps apply to both unicast and multicast traffic.*** To use a route map for unicast or multicast traffic only, you must use a match statement to specify the NLRI type.

Matching on Multicast Routes

To configure a route map to match on multicast routes only, enter commands such as the following:

```
BigIron(config)# access-list 10 permit 207.95.22.0 0.0.0.255
BigIron(config)# route-map mbgpmap permit 1
BigIron(config-routemap mbgpmap)# match nlri multicast
BigIron(config-routemap mbgpmap)# match ip address 10
```

These commands configure a route map that permits multicast routes (routes with NLRI type multicast) with source IP addresses that match the addresses specified in ACL 10.

Syntax: match nlri multicast | unicast | multicast unicast

Setting the NLRI Type to Multicast

To configure a route map to place routes that are permitted by the map into the multicast RIB but not the unicast RIB, enter commands such as the following:

```
BigIron(config)# access-list 20 permit 207.85.10.0 0.0.0.255
BigIron(config)# route-map mbgpmap2 permit 1
BigIron(config-routemap mbgpmap2)# match ip address 20
BigIron(config-routemap mbgpmap2)# set nlri multicast
```

These commands configure a route map that places all routes received from 207.85.10.x into the multicast RIB.

Syntax: set nlri multicast | unicast | multicast unicast

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. For an example, see “Enabling Redistribution of Directly-Connected Multicast Routes into MBGP” on page 17-6.

Additional Example

The following example shows two route map entries. These entries illustrate the need to explicitly specify the NLRI type (unicast or multicast) unless you want the route map entry to apply to both types. Beginning with software release 07.5.00, route maps entries match on both NLRI types by default.

```
BigIron(config)# route-map mbgpmapfoo permit 1
BigIron(config-routemap mbgpmapfoo)# match nlri unicast
BigIron(config-routemap mbgpmapfoo)# set med 20
BigIron(config-routemap mbgpmapfoo)# exit
BigIron(config)# route-map mbgpmapfoo permit 2
BigIron(config-routemap mbgpmapfoo)# match nlri multicast
BigIron(config-routemap mbgpmapfoo)# set med 50
```

The first route map entry (entry 1) matches on NLRI type unicast and sets the MED of matching routes to 20. The second entry (2) matches on NLRI type multicast and sets the MED of matching routes to 50.

Displaying MBGP Information

All of the BGP show commands have MBGP equivalents. Use **mbgp** instead of **bgp** in the command syntax. For example, to display the MBGP route table, enter the **show ip mbgp routes** command instead of the **show ip bgp routes** command. Table 17.1 lists the MBGP show commands and describes their output. For information about a command, see “Configuring BGP4” on page 16-1.

Table 17.1: MBGP Show Commands

Command	Description
show ip mbgp summary	Displays summary configuration information and statistics.
show ip mbgp config	Shows the configuration commands in the running-config.

Table 17.1: MBGP Show Commands (Continued)

Command	Description
show ip mbgp neighbors	Displays information about MBGP neighbors.
show ip mbgp peer-group	Displays information about MBGP peer groups.
show ip mbgp routes	Displays MBGP routes.
show ip mbgp <ip-addr>[/<prefix>]	Displays a specific MBGP route.
show ip mbgp attribute-entries	Displays MBGP route attributes.
show ip mbgp dampened-paths	Displays MBGP paths that have been dampened by route flap dampening.
show ip mbgp flap-statistics	Displays route flap dampening statistics.
show ip mbgp filtered-routes	Displays routes that have been filtered out.

The following sections show examples of some of the MBGP show commands. An example of the **show ip mroute** command is also included. This command displays the IP multicast route table.

Displaying Summary MBGP Information

To display summary MBGP information, enter the following command at any CLI prompt:

```
BigIron# show ip mbgp summary
  BGP4 Summary
  Router ID: 10.8.20.1   Local AS Number : 20
  Confederation Identifier : not configured
  Confederation Peers:
  Maximum Number of Paths Supported for Load Sharing : 1
  Number of Neighbors Configured : 6
  Number of Routes Installed : 3389
  Number of Routes Advertising to All Neighbors : 16936
  Number of Attribute Entries Installed : 750
  Neighbor Address  AS#   State   Time     Rt:Accepted Filtered Sent   ToSend
  1.1.1.1           2     CONN    0h 0m18s    0        0        0      3387
  7.7.7.1           30    ADMDN   0h16m56s    0        0        0      3387
  8.8.8.1           40    CONN    0h 0m18s    0        0        0      3387
  10.8.20.6         20    CONN    0h 0m 9s    0        0        0      3387
  15.15.15.2        40    ESTAB   0h16m35s    0        0        3387   0
  38.38.38.1        65097 ESTAB   0h16m44s    3388     0        1      0
```

Syntax: show ip mbgp summary

NOTE: This command's display looks similar to the display for the **show ip bgp config** command. However, the **show ip mbgp config** command lists only the MBGP neighbors, whereas the show ip bgp config command lists only the BGP neighbors.

Displaying the Active MBGP Configuration

To display the active MBGP configuration information contained in the running-config without displaying the entire running-config, enter the following command at any level of the CLI:

```
BigIron# show ip mbgp config
Current BGP configuration:
router bgp
 aggregate-address  192.1.0.0 255.255.0.0
 aggregate-address  192.1.0.0 255.255.0.0 nlri unicast multicast
 aggregate-address  207.95.0.0 255.255.0.0 nlri unicast multicast
 aggregate-address  207.95.0.0 255.255.0.0 summary-only
 as-path-filter  20 permit .
 local-as 20
 neighbor nj peer-group nlri unicast multicast
 neighbor 7.7.7.1 remote-as 30 nlri unicast multicast
 neighbor 7.7.7.1 shutdown
 neighbor 15.15.15.2 remote-as 40 nlri unicast multicast
 neighbor 38.38.38.1 remote-as 65097 nlri unicast multicast
 neighbor 1.1.1.1 peer-group nj
 neighbor 1.1.1.1 remote-as 2 nlri unicast multicast
 neighbor 10.8.20.6 remote-as 20 nlri unicast multicast
 neighbor 10.8.20.6 update-source loopback 1
 neighbor 10.8.20.6 route-map out newlocal
 neighbor 8.8.8.1 remote-as 40 nlri unicast multicast
 network 162.162.162.0 255.255.255.0 nlri unicast multicast
 redistribute connected route-map setcon
end
```

Syntax: show ip mbgp config

NOTE: This command displays exactly the same information as the **show ip bgp config** command. Each command displays both the BGP and MBGP configuration commands that are in the running-config.

Displaying MBGP Neighbors

To view MBGP neighbor information including the values for all the configured parameters, enter the following command. This display is similar to the **show ip bgp neighbor** display but has additional fields that apply only to MBGP. These fields are shown in bold type in the example and are explained below.

NOTE: The display shows all the configured parameters for the neighbor. Only the parameters that have values different from their defaults are shown.

```
BigIron(config-bgp-router)# show ip mbgp neighbor 7.7.7.2
    Total number of BGP Neighbors: 6
1   IP Address: 1.1.1.1, AS: 2 (EBGP), RouterID: 0.0.0.0
    State: CONNECT, Time: 1h27m5s, KeepAliveTime: 60, HoldTime: 180
       PeerGroup: nj
    Messages:    Open    Update  KeepAlive Notification Refresh-Req
       Sent    : 0       0       0         0            0
       Received: 0       0       0         0            0
    Last Connection Reset Reason:Unknown
    Notification Sent:     Unspecified
    Notification Received: Unspecified
    Neighbor NLRI Negotiation:
      Peer configured for Unicast and Multicast Routes
    TCP Connection state: ESTABLISHED
       Byte Sent:   1346, Received: 1714918
       Local host:  7.7.7.1, Local  Port: 179
       Remote host: 7.7.7.2, Remote Port: 8179
       ISentSeq:      12122  SendNext:      13469  TotUnAck:          0
       TotSent:        1347  ReTrans:           0  UnAckSeq:      13469
       IRcvSeq:   886310126  RcvNext:   888025045  SendWnd:       16384
       TotalRcv:    1714919  DupliRcv:        601  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. 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 Neighbor NLRI Negotiation section (shown in bold type) lists the types of routes that this Layer 3 Switch can exchange with the MBGP neighbor.

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 mbgp neighbors [<ip-addr>]

The <ip-addr> parameter specifies the neighbor's IP address.

Displaying MBGP Routes

To display the MBGP route table, enter the following command:

```
BigIron(config-bgp-router)# show ip mbgp routes
Total number of BGP Routes: 3389
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      10.10.2.0/24       38.38.38.1                 100        0      BE
         AS_PATH: 65097 65356
2      12.0.0.0/8         38.38.38.1                 100        0      BE
         AS_PATH: 65097 683 6509 24 1800 1239 5511 5511 2200 1305
3      12.4.125.0/24      38.38.38.1                 100        0      BE
         AS_PATH: 65097 683 11537 1239
4      12.6.92.0/24       38.38.38.1                 100        0      BE
         AS_PATH: 65097 683 11537 1239
5      12.150.219.0/24    38.38.38.1                 100        0      BE
         AS_PATH: 65097 683 11537 1239
6      24.144.0.0/18      38.38.38.1                 100        0      BE
         AS_PATH: 65097 683 11537 1239
7      24.221.128.0/19    38.38.38.1                 100        0      BE
         AS_PATH: 65097 683 11537 1239
8      24.221.160.0/19    38.38.38.1                 100        0      BE
         AS_PATH: 65097 683 11537 1239
9      35.0.0.0/8         38.38.38.1                 100        0      BE
         AS_PATH: 65097 683 11537 237
```

Syntax: show ip mbgp routes

Displaying the IP Multicast Route Table

To display the IP multicast route table, enter the following command:

```
BigIron(config-bgp-router)# show ip mroute
Total number of Mroutes: 3389
Start index: 1  D:Connected  R:RIP  S:Static  O:OSPF *:Candidate default
0    10.10.2.0        255.255.255.0     38.38.38.1       e3/5     20
1    12.0.0.0         255.0.0.0         38.38.38.1       e3/5     20
2    12.4.125.0       255.255.255.0     38.38.38.1       e3/5     20
3    12.6.92.0        255.255.255.0     38.38.38.1       e3/5     20
4    12.150.219.0     255.255.255.0     38.38.38.1       e3/5     20
5    24.144.0.0       255.255.192.0     38.38.38.1       e3/5     20
6    24.221.128.0     255.255.224.0     38.38.38.1       e3/5     20
7    24.221.160.0     255.255.224.0     38.38.38.1       e3/5     20
8    35.0.0.0         255.0.0.0         38.38.38.1       e3/5     20
9    62.4.0.0         255.255.224.0     38.38.38.1       e3/5     20
10   62.4.64.0        255.255.224.0     38.38.38.1       e3/5     20
11   62.8.32.0        255.255.224.0     38.38.38.1       e3/5     20
12   62.12.32.0       255.255.224.0     38.38.38.1       e3/5     20
13   62.16.32.0       255.255.224.0     38.38.38.1       e3/5     20
14   62.24.32.0       255.255.224.0     38.38.38.1       e3/5     20
15   62.41.0.0        255.255.0.0       38.38.38.1       e3/5     20
16   62.64.128.0      255.255.128.0     38.38.38.1       e3/5     20
17   62.67.0.0        255.255.0.0       38.38.38.1       e3/5     20
18   62.104.0.0       255.255.0.0       38.38.38.1       e3/5     20
19   62.104.192.0     255.255.224.0     38.38.38.1       e3/5     20
```

Syntax: show ip mroute

Chapter 18
Network Address Translation

You can configure a Foundry Layer 3 Switch to perform standard ***Network Address Translation (NAT)***. The following types of NAT are supported:

- Inside source NAT – Enables private IP networks that use nonregistered IP addresses to connect to the Internet.
- Inside destination NAT – Enables you to translate the global (Internet) IP addresses of traffic received from those addresses into private addresses.

NOTE: If you want to use NAT and ACLs on the same port, see "Using ACLs and NAT on the Same Interface (Flow-Based ACLs)" on page 6-65 for important guidelines.

Protocols Supported for NAT

Foundry NAT supports the following protocols:

- ICMP
- UDP/TCP (generic)
- FTP
- VDOLive
- StreamWorks
- CU-SeeMe
- RealAudio and RealVideo
- RealMedia
- QuickTime
- Microsoft Media Services
- Web Theater (Vxtreme)

NOTE: Foundry does not support streaming protocols, such as RTSP/MMS, if IP NAT inside destination static is configured.

NOTE: When configured for inside destination NAT, the Foundry device does not translate ICMP echo request packets from outside addresses to inside hosts. Instead, the device itself replies to the ping requests. The device does translate other types of ICMP packets.

Port Address Translation

Normally, NAT maps each address that needs to be translated to a unique IP address from a pool. However, it is possible for the address pool to have fewer addresses than the number of addresses you might need. In this case, you can configure the Foundry device to use Port Address Translation. ***Port Address Translation*** maps a client's IP address and TCP or UDP port number to both an IP address and a TCP or UDP port number. In this way, the Foundry device can map many addresses to the same address and use TCP or UDP port numbers to uniquely identify the private hosts.

NOTE: This type of feature is sometimes called OverloadingPort Overload.

In the example in Figure 18.1, a pool configured for inside source NAT contains enough addresses to ensure that every host on the private network can be mapped to an Internet address in the pool. However, suppose the enterprise implementing this configuration has only 20 Internet addresses. For example, the pool might be 209.157.1.1/24 – 209.157.1.20/24. In this case, the pool does not contain enough addresses to ensure that all the hosts in the private network can be mapped to Internet addresses.

Without Port Address Translation, it is possible that the device will not be able to provide NAT for some hosts. However, with Port Address Translation, the device can provide NAT for all the hosts by using a unique TCP or UDP port number in addition to the IP address to map to each host. For example, the device can map the following addresses:

Inside address	Outside address
10.10.10.2:6000	209.157.1.2:4000
10.10.10.3:6000	209.157.1.2:4001
10.10.10.4:6000	209.157.1.2:4002

NAT is mapping the same global IP address to three different private addresses along with their TCP or UDP ports, but uses a different TCP or UDP port number for each private address to distinguish them. Notice that the Port Address Translation feature does not attempt to use the same TCP or UDP port number as in the client's packet.

The way NAT deals with the client's TCP or UDP port number depends on whether Port Address Translation is enabled:

- Port Address Translation enabled – NAT treats the client's IP address and TCP or UDP port number as a single entity, and uniquely maps that entity to another entity consisting of an IP address and TCP or UDP port number. The NAT entry the device creates in the NAT translation table therefore consists of an IP address plus a TCP or UDP port number. The device maintains the port type in the translation address:
  - If the client's packet contains a TCP port number, the device uses a TCP port in the translation address.
  - If the client's packet contains a UDP port, the device uses a UDP port in the translation address.

  The device does not try to use the same TCP or UDP port number for the untranslated and translated addresses. Instead, the device maps the client IP address plus the TCP or UDP port number to a unique combination of IP address plus TCP or UDP port number. When the device receives reply traffic to one of these hosts, NAT can properly translate the Internet address back into the private address because the TCP or UDP port number in the translation address uniquely identifies the host.

To enable Port Address Translation, use the overload option when you configure the source list, which associates a private address range with a pool of Internet addresses. See “Configuring Dynamic NAT Parameters” on page 18-6.

- Port Address Translation disabled – The device translates only the client’s IP address into another IP address and retains the TCP or UDP port number unchanged.

Maximum Number of Addresses

If the Layer 3 Switch cannot allocate an address because it has run out of addresses, the Layer 3 Switch drops the packet and sends an ICMP Host Unreachable packet.

NOTE: The maximum number of global IP addresses you can configure for inside source NAT depends on how much memory the Layer 3 Switch has and whether you enable the Port Address Translation feature. Regardless of the amount of memory, you cannot configure more than 256 global IP addresses.

Inside Source NAT

Inside source NAT enables private IP networks that use nonregistered IP addresses to connect to the Internet. Configure the NAT on the Foundry device at the border of an inside network and an outside network (such as the Internet). Inside source NAT translates the internal local addresses to globally unique IP addresses before sending packets to the outside network. Inside source NAT also allows a more graceful renumbering strategy for organizations that are changing service providers or voluntarily renumbering into Classless Interdomain Routing (CIDR) blocks.

Use inside source NAT to translate your private (inside) IP addresses into globally unique (outside) IP addresses when communicating outside of your network.

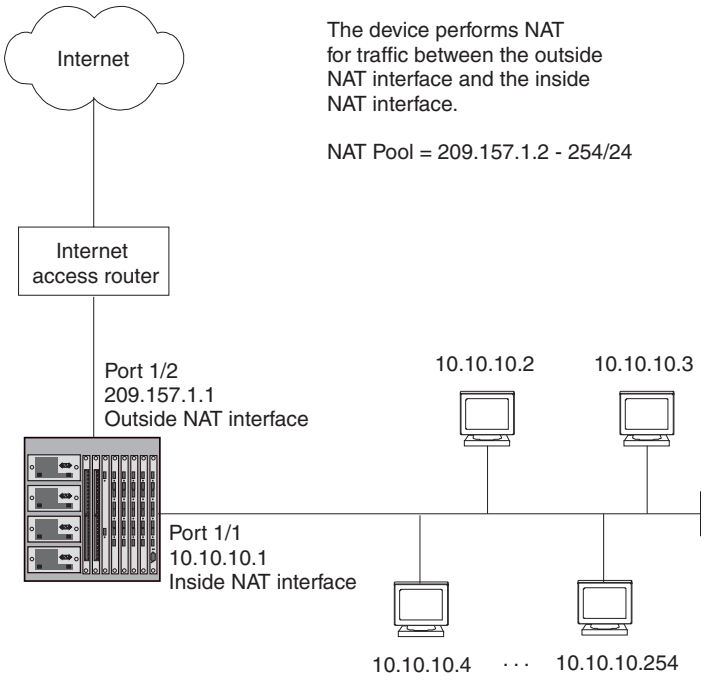
NOTE: This feature is supported on all chassis Layer 3 Switches with Management 2 modules or higher and on the IP-only NetIron Stackable Layer 3 Switch.

NOTE: The maximum number of global IP addresses you can configure depends on how much memory the Layer 3 Switch has and whether you enable the Port Address Translation feature. Regardless of the amount of memory, you cannot configure more than 256 global IP addresses.

A Foundry device configured for NAT must have an interface to the private network and an interface to a public network (for example, the Internet). In a typical environment, NAT is configured on the Foundry device between the private network and the Internet. When you configure a Foundry device for NAT, the device does not advertise the private networks to the Internet. However, the device can advertise route information received from the Internet to the private networks.

Figure 18.1 shows a basic example of a network using NAT on a Foundry device. In this example, a BigIron 8000 Layer 3 Switch is using NAT to translate traffic originated from the hosts on the 10.10.10.x/24 subnet into public addresses from the address pool.

Figure 18.1 Network Using Inside NAT



In this example, the Layer 3 Switch is configured to perform dynamic NAT to translate between the private addresses in the 10.10.10.x/24 subnet and the Internet addresses in the 209.157.1.x/24 subnet.

NOTE: This example is simplified to show how NAT is used. For detailed configuration examples, see "Configuration Examples" on page 18-8.

To configure NAT on a Layer 3 Switch, you must configure an inside NAT interface and an outside NAT interface.

- The inside NAT interface is connected to the private addresses.
- The outside NAT interface is connected to the Internet.

The inside NAT interface in Figure 18.1 uses the address pool 209.157.1.2/24 – 209.157.1.254/24 to map the private addresses to public addresses for traffic initiated by hosts in the 10.10.10.x/24 subnet.

You can configure the following types of NAT:

- Dynamic NAT – Dynamic NAT maps private addresses to public addresses in a pool. The public addresses come from a pool of addresses that you configure. In the example in Figure 18.1, the pool is the range of addresses from 209.157.1.2/24 – 209.157.1.254/24. When you use dynamic NAT, the software uses a round robin technique to select a global IP address to map to a private address from a pool that you configure.
- Static NAT – Static NAT maps one particular global IP address with one particular private address. Use static NAT when you want to ensure that the software always maps the same global address to a given private address. For example, use static NAT when you want specific hosts in the private network to always use the same Internet address when communicating outside the private network.

  If you want to a one to one mapping of addresses only from inside NAT to outside NAT (and not from the outside NAT to the inside NAT) use dynamic NAT with only one IP address in the NAT pool and in the NAT ACL.

  NOTE: Static and inside destination static NAT mappings are bi-directional.

NOTE: You can configure both dynamic and static NAT on the same Foundry device. When you configure both types of NAT, static NAT takes precedence over dynamic NAT. Thus, if you configure a static NAT translation for a private address, the device always uses that translation instead of creating a dynamic one.

Configuring Source NAT

To configure NAT, perform the following tasks:

- Configure the static address mappings, if needed. Static mappings explicitly map a specific private address to a specific Internet address to ensure that the addresses are always mapped together. Use static address mappings when you want to ensure that a specific host in the private network is always mapped to the Internet address you specify.
- Configure dynamic NAT parameters:
  - Configure a standard or extended ACL for each range of private addresses for which you want to provide NAT.
  - Configure a pool for each consecutive range of Internet addresses to which you want NAT to be able to map the private addresses specified in the ACLs. Each pool must contain a range with no gaps. If your Internet address space has gaps, configure separate pools for each consecutive range within the address space.
  - Associate a range of private addresses (specified in a standard or extended ACL) with a pool.
  - Optionally, enable the Port Address Translation feature. Use this feature if you have more private addresses that might need NAT than the Internet address pools contain.

NOTE: If you plan to use dynamic NAT without the Port Address Translation feature, contact your Foundry account representative for additional requirements that may apply to your installation.

- Enable inside NAT on the interface connected to the private addresses.
- Enable outside NAT on the interface connected to global addresses.

The configuration does not take effect until you enable inside and outside NAT on specific interfaces.

NOTE: You must configure inside NAT on one interface and outside NAT on another interface. The device performs NAT for traffic between the interfaces.

NOTE: Named ACLS are not supported with NAT. You must use a numbered ACL.

In addition to the tasks listed above, you can modify the age timers for the address translation entries the device creates. See “Changing Translation Table Timeouts” on page 18-16 for information. For information about viewing the active NAT translations, see “Displaying the Active NAT Translations” on page 18-17.

The following sections provide procedures for configuring NAT.

Configuring Static Address Translations

Use the following CLI method to configure static NAT.

USING THE CLI

To configure static NAT for an IP address, enter a command such as the following:

```
BigIron(config)# ip nat inside source static 10.10.10.69 209.157.1.69
```

The command in this example statically maps the private address 10.10.10.69 to the Internet address 209.157.1.69.

Syntax: [no] ip nat inside source static <private-ip> <global-ip>

This command associates a specific private address with a specific Internet address. Use this command when you want to ensure that the specified addresses are always mapped together.

The **inside source** parameter specifies that the mapping applies to the private address sending traffic to the Internet.

The <private-ip> parameter specifies the private IP address.

The <global-ip> parameter specifies the Internet address. The device supports up to 256 global IP addresses.

Neither of the IP address parameters needs a network mask.

Configuring Dynamic NAT Parameters

To configure dynamic NAT:

- Configure a standard or extended ACL for each private address range.

NOTE: Named ACLS are not supported with NAT. You must use a numbered ACL.

- Configure a pool for each consecutive range of Internet addresses.
- Associate private addresses (ACLs) with pools.
- Optionally, enable the Port Address Translation feature.

NOTE: If you plan to use dynamic NAT without the Port Address Translation feature, contact your Foundry account representative for additional requirements that may apply to your installation.

Use the following CLI method to configure dynamic NAT.

USING THE CLI

You can configure dynamic NAT with the Port Address Translation feature disabled or enabled.

Example with Port Address Translation Disabled

NOTE: If you plan to use dynamic NAT without the Port Address Translation feature, contact your Foundry account representative for additional requirements that may apply to your installation.

To configure dynamic NAT with the Port Address Translation feature disabled, enter commands such as the following at the global CONFIG level of the CLI:

```
BigIron(config)# access-list 1 permit 10.10.10.0/24
BigIron(config)# ip nat pool OutAdds 209.157.1.2 209.157.1.254 prefix-length 24
BigIron(config)# ip nat inside source list 1 pool OutAdds
```

These commands configure a standard ACL for the private subnet 10.10.10.x/24, then enable inside NAT for the subnet. Make sure you specify **permit** in the ACL, rather than **deny**. If you specify **deny**, the Foundry device will not provide NAT for the addresses.

Example with Port Address Translation Enabled

To configure dynamic NAT with the Port Address Translation feature enabled, enter commands such as the following at the global CONFIG level of the CLI:

```
BigIron(config)# access-list 1 permit 10.10.10.0/24
BigIron(config)# ip nat pool OutAdds 209.157.1.2 209.157.1.254 prefix-length 24
BigIron(config)# ip nat inside source list 1 pool OutAdds overload
```

These commands are the same as the ones in “Example with Port Address Translation Disabled”, except the **ip nat inside source** command uses the **overload** parameter. This parameter enables the Port Address Translation feature.

Command Syntax

Syntax: [no] ip nat pool <pool-name> <start-ip> <end-ip> netmask <ip-mask> | prefix-length <length> [type match-host | rotary]

This command configures the address pool.

The <pool-name> parameter specifies the pool name. The name can be up to 255 characters long and can contain special characters and internal blanks. If you use internal blanks, you must use quotation marks around the entire name.

The <start-ip> parameter specifies the IP address at the beginning of the pool range. Specify the lowest-numbered IP address in the range.

The <end-ip> parameter specifies the IP address at the end of the pool range. Specify the highest-numbered IP address in the range.

NOTE: The address range cannot contain any gaps. Make sure you own all the IP addresses in the range. If the range contains gaps, you must create separate pools containing only the addresses you own.

The **netmask** <ip-mask> | **prefix-length** <length> parameter specifies a classical subnet mask (example: **netmask** 255.255.255.0) or the length of a Classless Interdomain Routing prefix (example: **prefix-length 24**).

NOTE: The maximum number of global IP addresses you can configure depends on how much memory the Layer 3 Switch has and whether you enable the Port Address Translation feature. Regardless of the amount of memory, you cannot configure more than 256 global IP addresses.

The **type match-host | rotary** parameter specifies the method the software uses to assign the host portion of the translated address.

- **match-host** – The software uses the same host address as the untranslated address. For example, if the untranslated address is 192.2.4.69 and the host portion of the address is 69, the translated address also uses the host address 69. This method results in the translated addresses always having the same host addresses as their untranslated counterparts.
- **rotary** – The software assigns a host address from 1 – 254, beginning with 1 for the first translated address. This is the default.

Syntax: [no] ip nat inside source list <acl-id> pool <pool-name> [overload]

This command associates a private address range with a pool of Internet addresses and optionally enables the Port Address Translation feature.

The **inside source** parameter specifies that the translation applies to private addresses sending traffic to global addresses (Internet addresses).

The **list** <acl-id> parameter specifies a standard or extended ACL. You can specify a numbered or named ACL.

NOTE: Named ACLS are not supported with NAT. You must use a numbered ACL.

NOTE: For complete standard and extended ACL syntax, see “Access Control List” on page 6-1.

The **pool** <pool-name> parameter specifies the pool. You must create the pool before you can use it with this command.

The **overload** parameter enables the Port Address Translation feature. Use this parameter if the IP address pool does not contain enough addresses to ensure NAT for each private address. The Port Address Translation feature conserves Internet addresses by mapping the same Internet address to more than one private address and using a TCP or UDP port number to distinguish among the private hosts. The device supports up to 50 global IP addresses with this feature enabled.

Enabling NAT

The NAT configuration does not take effect until you enable it on specific interfaces. You can enable NAT on Ethernet ports and on virtual interfaces. You also can enable the feature on the primary port of a trunk group, in which case the feature applies to all the ports in the trunk group.

NOTE: You must configure inside NAT on one interface and outside NAT on another interface. The device performs NAT for traffic between the interfaces.

To enable NAT, use the following CLI methods.

Enabling Inside NAT

To enable inside NAT on the interface attached to the private addresses, use the following CLI method.

USING THE CLI

To enable inside NAT on an interface, enter commands such as the following:

```
BigIron(config)# interface ethernet 1/1
BigIron(config-if-1/1)# ip nat inside
```

This command enables inside NAT on Ethernet port 1/1.

Syntax: [no] ip nat inside

To enable inside NAT on a virtual interface, enter commands such as the following:

```
BigIron(config)# interface ve 1
BigIron(config-vif-1)# ip nat inside
```

This command enables inside NAT on virtual interface 4.

Enabling Outside NAT

To enable outside NAT on the interface attached to public addresses, use the following CLI method.

USING THE CLI

To enable outside NAT on an interface, enter commands such as the following:

```
BigIron(config)# interface ethernet 1/2
BigIron(config-if-1/2)# ip nat outside
```

This command enables outside NAT on Ethernet port 1/2.

Syntax: [no] ip nat outside

To enable outside NAT on a virtual interface, enter commands such as the following:

```
BigIron(config)# interface ve 2
BigIron(config-vif-2)# ip nat outside
```

This command enables outside NAT on virtual interface 4.

Configuration Examples

This section shows two complete configuration examples for NAT. The examples are based on different network topologies.

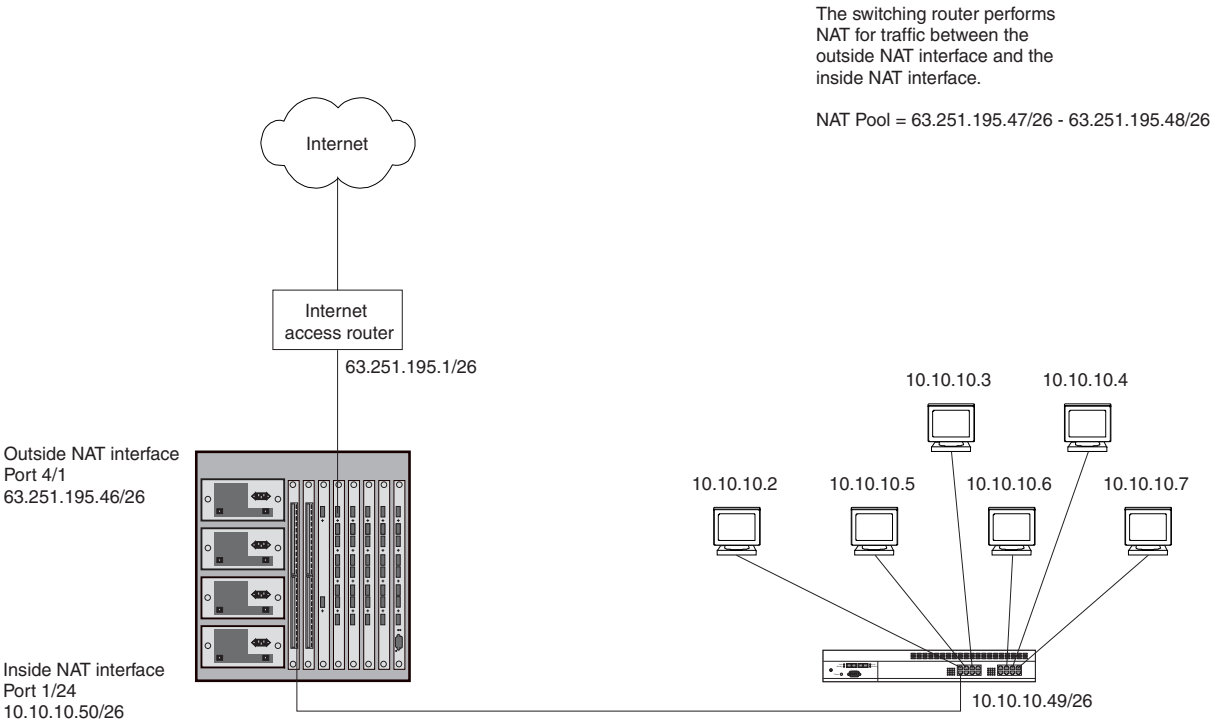
- NAT clients connected to the Layer 3 Switch by a Switch.
- NAT clients connected directly to Layer 3 Switch ports.

NOTE: You also can enable the feature on the primary port of a trunk group, in which case the feature applies to all the ports in the trunk group. These examples do not show this configuration.

Private NAT Clients Connected to the Layer 3 Switch by a Switch

Figure 18.2 shows an example of a NAT configuration in which the clients in the private network are attached to the Layer 3 Switch through a Switch.

Figure 18.2 NAT clients connected to the Layer 3 Switch by a Switch



Here are the CLI commands for implementing the NAT configuration shown in Figure 18.3. These commands configure the following:

- An IP address and default gateway on the Layer 2 Switch
- An Access Control List (ACL) for the range of private addresses in the private network on virtual interface 10
- A Pool of public (Internet) address to use for translation of the private addresses
- An association of the ACL for the private addresses with the pool for translation
- A default route that has the Internet access router as the route's next-hop gateway

The commands also enable inside NAT and outside NAT on the ports connected to the private network's Switch and to the Internet access router, and save the configuration changes to the startup-config file.

Layer 2 Switch Commands

The following commands access the configuration level of the CLI on the Foundry FastIron Workgroup Layer 2 Switch, then configure an IP address and specify the default gateway. The Layer 2 Switch connects the private address clients to the Layer 3 Switch in Figure 18.2. The default gateway is the Layer 3 Switch's IP interface with the Layer 2 Switch.

```
FastIron> en
FastIron# configure terminal
FastIron(config)# ip address 10.10.10.49/26
FastIron(config)# ip default-gateway 10.10.10.50/26
```

The following command saves the configuration to the Layer 2 Switch's startup-config file on flash memory. The Layer 2 Switch applies the configuration information as soon as you enter it into the CLI. Saving the changes to the startup-config file ensures that the changes are reinstated following a system reload.

```
FastIron(config)# write memory
```


Layer 3 Switch Commands

The following commands access the configuration level of the CLI.

```
BigIron> en
BigIron# configure terminal
BigIron(config)#
```

The following command configures an ACL to identify the range of private addresses for which you want to provide NAT services. This ACL identifies the private address range as 10.10.10.0 – 10.10.10.255.

```
BigIron(config)# access-list 9 permit 10.10.10.0 0.0.0.255
```

NOTE: The format of the network mask for an ACL uses zeroes to indicate a value that must match, and ones (255 in decimal) as a wildcard. In this case, 0.0.0.255 means the first three parts of the IP address must match exactly, but the fourth part can have any value.

The following command configures the NAT address pool. The Layer 3 Switch translates a client's address from the private network to an address from this pool when the client sends traffic to a public network, in this case a network located somewhere on the Internet.

```
BigIron(config)# ip nat pool np1 63.251.195.47 63.251.195.48 netmask 255.255.255.192
```

This command configures a pool named "np1", and adds public address range 63.251.195.47/26 – 63.251.195.48/26 to the pool. Generally, a pool contains more than two addresses, but this pool is small so that this configuration can also demonstrate the Port Address Translation feature.

The following command associates the range of private addresses identified by the ACL with the pool, and in this case also enables the Port Address Translation feature. Port Address Translation allows you to use an address pool that contains fewer addresses than the number of NAT clients in the private network.

```
BigIron(config)# ip nat inside source list 9 pool np1 overload
```

The **inside source list 9** portion of the command identifies the range of source addresses. The value "9" is the number of the ACL configured above. The **pool np1** portion of the command identifies the IP address pool configured above. The **overload** parameter enables Port Address Translation. When this feature is enabled, NAT associates a TCP or UDP port number with the public address for a client. In this case, there are four clients but only two addresses in the pool. Port Address Translation allows NAT to provide translation addresses for all four clients. When two translation clients have the same public IP address, the software can still distinguish between the clients because each client has a unique TCP or UDP port number.

The following command configures a static default route to the Internet access router. The Layer 3 Switch uses this route for traffic that is addressed to a destination for which the IP route table does not have an explicit route. Typically, the IP route table does not have explicit routes to all destination networks on the Internet.

```
BigIron(config)# ip route 0.0.0.0 0.0.0.0 63.251.195.1
```

The address 0.0.0.0 0.0.0.0 is the standard notation for an IP default route. The 63.251.195.1 address is the address of the next-hop gateway for the route. In this case, the next-hop gateway is the Layer 3 Switch's IP interface with Internet access router.

The following commands change to the configuration level for port 1/24, configure an IP address on the port, and enable inside NAT on the port. Port 1/24 connects the Layer 3 Switch to the Switch, which is connected to the private network containing the NAT clients.

```
BigIron(config)# interface ethernet 1/24
BigIron(config-if-1/24)# ip address 10.10.10.50 255.255.255.192
BigIron(config-if-1/24)# ip nat inside
BigIron(config-if-1/24)# exit
```

The following commands change to the configuration level for port 4/1, configure an IP address on the port, and enable outside NAT on the port. Port 4/1 connects the Layer 3 Switch to the Internet access device.

```
BigIron(config)# interface ethernet 4/1
BigIron(config-if-4/1)# ip address 63.251.195.46 255.255.255.192
BigIron(config-if-4/1)# ip nat outside
```



```
BigIron(config-if-4/1)# exit
```

The following command saves all the configuration changes above to the Layer 3 Switch's startup-config file on flash memory. The Layer 3 Switch applies NAT configuration information as soon as you enter it into the CLI. Saving the changes to the startup-config file ensures that the changes are reinstated following a system reload.

```
BigIron(config)# write memory
```

Private NAT Clients Connected Directly to the Layer 3 Switch

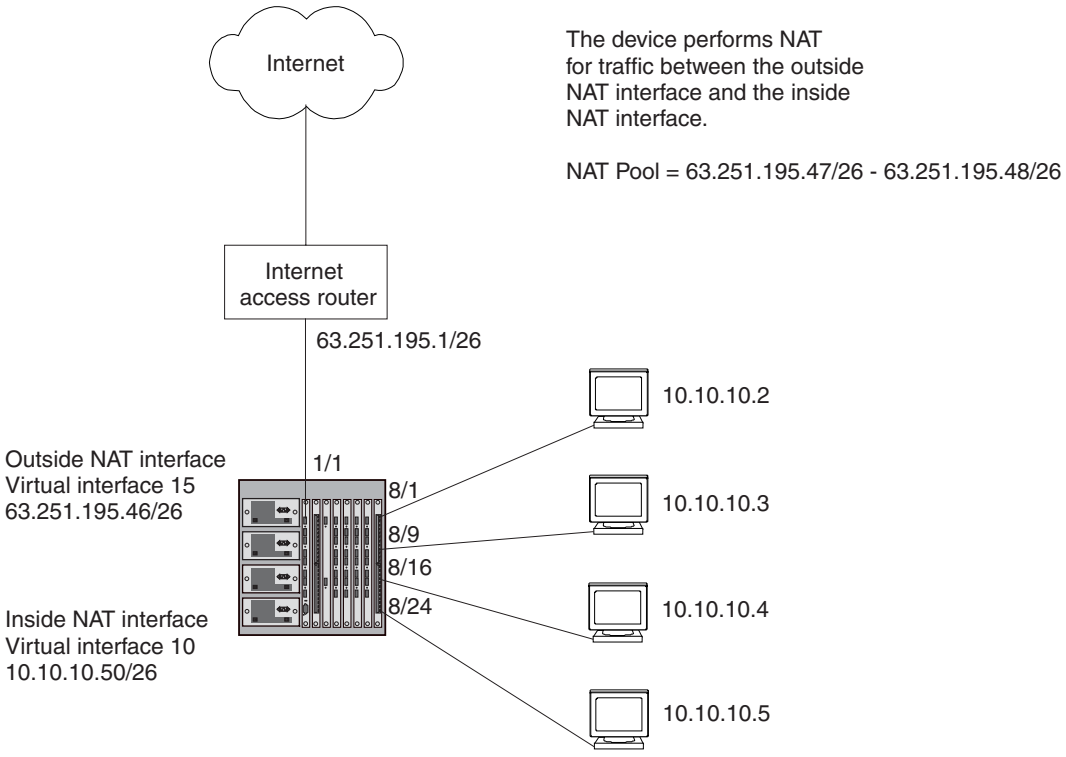
Figure 18.3 shows an example of a NAT configuration in which the NAT clients on the private network are directly connected to the Layer 3 Switch. The configuration commands are similar to those for the configuration in "Private NAT Clients Connected to the Layer 3 Switch by a Switch" on page 18-8, except the inside NAT and outside NAT interfaces are virtual routing interfaces (called virtual interfaces or "VEs") instead of physical ports.

Since all the clients are in the same subnet, the Layer 3 Switch is configured with a virtual interface to serve as the inside NAT interface, the Layer 3 Switch's IP interface for the NAT clients who have private addresses.

The virtual interface is required because you cannot configure IP addresses in the same subnet on multiple physical interfaces on the Layer 3 Switch. A virtual interface is a logical interface that allows you to associate the same IP address (the IP address of the virtual interface) with multiple physical ports.

You can use a virtual interface for routing only when you add the interface to a port-based VLAN. A port-based VLAN is a separate Layer 2 broadcast domain, a logical Switch within the Foundry device. The Layer 3 Switch uses virtual interfaces to route Layer 3 traffic between port-based VLANs. Thus, this configuration also includes configuration of separate port-based VLANs for the clients' inside NAT interface and for the outside NAT interface.

Figure 18.3 NAT clients connected directly to the Layer 3 Switch



Here are the CLI commands for implementing the NAT configuration shown in Figure 18.3. These commands configure the following:

- Port-based VLAN 2 and virtual interface 10 for the inside NAT interface
- Port-based VLAN 3 and virtual interface 15 for the outside NAT interface
- An Access Control List (ACL) for the range of private address in the private network on virtual interface 10

- A Pool of public (Internet) address to use for translation of the private addresses
- An association of the ACL for the private addresses with the pool for translation
- A default route that has the Internet access router as the route's next-hop gateway

The commands also enable inside NAT and outside NAT on the virtual interfaces and save the configuration changes to the startup-config file. All the commands are entered on the Layer 3 Switch.

The following commands access the configuration level of the CLI, then configure port-based VLAN 2 and add virtual interface 10 to the VLAN.

```
BigIron> en
BigIron# configure terminal
BigIron(config)# vlan 2 by port
BigIron(config-vlan-2)# untagged ethernet 8/1 to 8/24
BigIron(config-vlan-2)# router-interface ve 10
BigIron(config-vlan-2)# exit
```

These commands add ports 8/1 through 8/24 as untagged ports to port-based VLAN 2. Generally, unless a port is a member of more than one port-based VLAN, you do not need to tag the port. The **router-interface 10** command adds virtual interface 10. At this point the virtual interface does not have an IP address associated with it.

The following commands add port-based VLAN 3 and add virtual interface 15 to the VLAN.

```
BigIron(config)# vlan 3 by port
BigIron(config-vlan-3)# untagged ethernet 1/1
BigIron(config-vlan-3)# router-interface ve 15
BigIron(config-vlan-3)# exit
```

The following command configures an ACL to identify the range of private addresses for which you want to provide NAT services. This ACL identifies the private address range as 10.10.10.0 – 10.10.10.255.

```
BigIron(config)# access-list 9 permit 10.10.10.0 0.0.0.255
```

NOTE: The format of the network mask for an ACL uses zeroes to indicate a value that must match, and ones (255 in decimal) as a wildcard. In this case, 0.0.0.255 means the first three parts of the IP address must match exactly, but the fourth part can have any value.

The following command configures the NAT address pool. The Layer 3 Switch translates a client's address from the private network to an address from this pool when the client sends traffic to a public network, in this case a network located somewhere on the Internet.

```
BigIron(config)# ip nat pool np1 63.251.195.47 63.251.195.48 netmask 255.255.255.192
```

This command configures a pool named "np1", and adds public address range 63.251.195.47/26 – 63.251.195.48/26 to the pool. Generally, a pool contains more than two addresses, but this pool is small so that this configuration can also demonstrate the Port Address Translation feature.

The following command associates the range of private addresses identified by the ACL with the pool, and in this case also enables the Port Address Translation feature. Port Address Translation allows you to use an address pool that contains fewer addresses than the number of NAT clients in the private network.

```
BigIron(config)# ip nat inside source list 9 pool np1 overload
```

The **inside source list 9** portion of the command identifies the range of source addresses. The value "9" is the number of the ACL configured above. The **pool np1** portion of the command identifies the IP address pool configured above. The **overload** parameter enables Port Address Translation. When this feature is enabled, NAT associates a TCP or UDP port number with the public address for a client. In this case, there are four clients but only two addresses in the pool. Port Address Translation allows NAT to provide translation addresses for all four clients. When two translation clients have the same public IP address, the software can still distinguish between the clients because each client has a unique TCP or UDP port number.

The following command configures a static default route to the Internet access router. The Layer 3 Switch uses this route for traffic that is addressed to a destination for which the IP route table does not have an explicit route. Typically, the IP route table does not have explicit routes to all destination networks on the Internet.

```
BigIron(config)# ip route 0.0.0.0 0.0.0.0 63.251.195.1
```

The address 0.0.0.0 0.0.0.0 is the standard notation for an IP default route. The 63.251.195.1 address is the address of the next-hop gateway for the route. In this case, the next-hop gateway is the Layer 3 Switch's IP interface with Internet access router.

The following commands configure an IP address on virtual interface 10, which is the virtual interface for the private network, and enable inside NAT on the interface.

```
BigIron(config)# interface ve 10
BigIron(config-ve-10)# ip address 10.10.10.50 255.255.255.192
BigIron(config-ve-10)# ip nat inside
BigIron(config-ve-10)# exit
```

The following commands configure an IP address on virtual interface 15, which is the interface to the Internet access router, and enable outside NAT on the interface.

```
BigIron(config)# interface ve 15
BigIron(config-ve-15)# ip address 63.251.195.46 255.255.255.192
BigIron(config-ve-15)# ip nat outside
BigIron(config-ve-15)# exit
```

The following command saves all the configuration changes above to the Layer 3 Switch's startup-config file on flash memory. The Layer 3 Switch applies NAT configuration information as soon as you enter it into the CLI. Saving the changes to the startup-config file ensures that the changes are reinstated following a system reload.

```
BigIron(config)# write memory
```

Inside Destination NAT

Inside destination NAT translates the global (Internet) IP addresses of traffic received from those addresses into private addresses. You can use the feature to associate an application, such as email, with a specific internal host. The feature also can provide load balancing for services by mapping traffic addressed to the services to a pool of internal addresses, and thus to different hosts.

Use the feature on a Layer 3 Switch that has at least one interface to the private network and another interface to the Internet. For example, use this feature on a Layer 3 Switch that connects a stub domain to the backbone

You can configure either of the following types of inside destination NAT:

- Dynamic NAT – Dynamic NAT maps global addresses to private addresses, which are in a pool that you configure. When you use dynamic NAT, the software uses a round robin technique to select the private addresses from the pool.
- Static NAT – Static NAT maps a particular private address to a particular global address. Use static NAT when you want to ensure that the software always maps the same private address to a given global address. Optionally, you also can map traffic based on TCP or UDP application ports, to provide NAT for specific applications.

NOTE: You can configure both dynamic and static inside destination NAT on the same Foundry device. When you configure both types of NAT, static NAT takes precedence over dynamic NAT. If you configure a static NAT translation for an address, the device always uses that translation instead of creating a dynamic one.

Configuring Inside Destination NAT

To configure inside destination NAT:

- Configure the static address mappings, if needed. Static mappings explicitly map a specific public address to a specific private address to ensure that the addresses are always mapped together. Use static address mappings when you want to ensure that a specific public host is always mapped to a specific private host.
- Configure dynamic NAT parameters:
  - Configure an IP ACL (standard or extended) for each range of public addresses for which you want to provide NAT.
  - Configure a pool for each consecutive range of private addresses to which you want NAT to be able to map the public addresses specified in the ACLs. Each pool must contain a range with no gaps. If your private address space has gaps, configure separate pools for each consecutive range within the address space.
  - Associate a range of public addresses (specified in an IP ACL) with a pool.
- Enable inside destination NAT on the interface connected to the private addresses.
- Enable outside NAT on the interface connected to global addresses.

The configuration does not take effect until you enable inside destination and outside NAT on specific interfaces.

Configuring Static Inside Destination NAT for IP Addresses Only

To configure static inside destination NAT for an IP address, enter a command such as the following:

```
BigIron(config)# ip nat inside destination static 209.157.1.69 10.10.10.69
```

The command in this example statically maps the Internet address 209.157.1.69 to the private address 10.10.10.69.

Syntax: [no] ip nat inside destination static <global-ip> <private-ip>

The **inside destination** parameter specifies that the mapping applies to the Internet address sending traffic to the private network.

The <global-ip> parameter specifies the Internet address.

The <private-ip> parameter specifies the private IP address.

Neither of the IP address parameters needs a network mask.

Configuring Static Inside Destination NAT for IP Addresses and TCP or UDP Ports

---

NOTE: When configuring static inside destination NAT by port, streaming media protocols (e.g., RTSP and MMS) and passive FTP are not supported.

To include TCP or UDP application port numbers in the translation, enter a command such as the following:

```
BigIron(config)# ip nat inside destination static tcp 209.157.1.69 80 10.10.10.69
8080
```

This command provides the same IP address translation as the previous command example. However, this command also translates TCP port 80 to TCP port 8080. The translation applies to the destination port, for inbound traffic.

Syntax: [no] ip nat inside destination static tcp | udp <global-ip> <global-tcp/udp> <private-ip> <private-tcp/udp>

The **tcp | udp** parameter indicates that you are creating a static mapping for a specify application (TCP or UDP port).

The <global-tcp/udp> parameter specifies the application port on the public host.

The <private-tcp/udp> parameter specifies the application port on the private host.

The other parameters are the same as described in “Configuring Static Inside Destination NAT for IP Addresses Only”.

Configuring Dynamic Inside Destination NAT

To configure dynamic inside destination NAT:

- Configure a standard or extended ACL for each public address range.
- Configure a pool for each consecutive range of private addresses.
- Associate public addresses (ACLs) with pools.

To configure dynamic inside destination NAT, enter commands such as the following at the global CONFIG level of the CLI:

```
BigIron(config)# access-list 1 permit 209.157.1.2/24
BigIron(config)# ip nat pool InAdds 10.10.10.0 10.10.10.254 prefix-length 24
BigIron(config)# ip nat inside destination list 1 pool InAdds
```

These commands configure a standard ACL for the public network 10.10.10.x/24, then enable inside destination NAT for the network. Make sure you specify **permit** in the ACL, rather than **deny**. If you specify **deny**, the Foundry device will not provide NAT for the addresses.

Syntax: [no] ip nat pool <pool-name> <start-ip> <end-ip> netmask <ip-mask> | prefix-length <length> [type match-host | rotary]

This command configures the address pool.

The <pool-name> parameter specifies the pool name. The name can be up to 255 characters long and can contain special characters and internal blanks. If you use internal blanks, you must use quotation marks around the entire name.

The <start-ip> parameter specifies the IP address at the beginning of the pool range. Specify the lowest-numbered IP address in the range.

The <end-ip> parameter specifies the IP address at the end of the pool range. Specify the highest-numbered IP address in the range.

NOTE: The address range cannot contain any gaps. Make sure you own all the IP addresses in the range. If the range contains gaps, you must create separate pools containing only the addresses you own.

The **netmask** <ip-mask> | **prefix-length** <length> parameter specifies a classical subnet mask (example: **netmask** 255.255.255.0) or the length of a Classless Interdomain Routing prefix (example: **prefix-length 24**).

The **type match-host | rotary** parameter specifies the method the software uses to assign the host portion of the translated address.

- **match-host** – The software uses the same host address as the untranslated address. For example, if the untranslated address is 192.2.4.69 and the host portion of the address is 69, the translated address also uses the host address 69. This method results in the translated addresses always having the same host addresses as their untranslated counterparts.
- **rotary** – The software assigns a host address from 1 – 254, beginning with 1 for the first translated address. This is the default.

Syntax: [no] ip nat inside destination list <acl-name-or-num> pool <pool-name>

This command associates a public address range with a pool of private addresses.

The **inside destination** parameter specifies that the translation applies to public addresses sending traffic to private addresses.

The **list** <acl-name-or-num> parameter specifies an IP ACL (standard or extended). You can specify a numbered or named ACL.

The **pool** <pool-name> parameter specifies the pool. You must create the pool before you can use it with this command.

Enabling NAT

The NAT configuration does not take effect until you enable it on specific interfaces. You can enable NAT on Ethernet ports and on virtual interfaces. You also can enable the feature on the primary port of a trunk group, in which case the feature applies to all the ports in the trunk group.

NOTE: You must configure inside destination NAT on one interface and outside NAT on another interface. The device performs NAT for traffic between the interfaces.

To enable inside destination NAT on the interface attached to the private addresses, enter commands such as the following:

```
BigIron(config)# interface ethernet 1/1
BigIron(config-if-1/1)# ip nat inside
```

This command enables inside destination NAT on Ethernet port 1/1.

Syntax: [no] ip nat inside

To enable inside destination NAT on a virtual interface, enter commands such as the following:

```
BigIron(config)# interface ve 1
BigIron(config-vif-1)# ip nat inside
```

This command enables inside destination NAT on virtual interface 4.

To enable outside NAT on an interface, enter commands such as the following:

```
BigIron(config)# interface ethernet 1/2
BigIron(config-if-1/2)# ip nat outside
```

This command enables outside NAT on Ethernet port 1/2.

Syntax: [no] ip nat outside

To enable outside NAT on a virtual interface, enter commands such as the following:

```
BigIron(config)# interface ve 2
BigIron(config-vif-2)# ip nat outside
```

This command enables outside NAT on virtual interface 4.

Changing Translation Table Timeouts

The NAT translation table contains all the currently active NAT translation entries on the device. NAT performs the following steps to provide an address translation for a source IP address:

- The feature looks in the NAT translation table for an active NAT entry for the translation. If the table contains an active entry for the session, the device uses that entry.
- If NAT does not find an active entry in the NAT translation table, NAT creates an entry and places the entry in the table. The entry remains in the table until the entry times out.

Each NAT entry remains in the NAT translation table until the entry ages out. The age timers apply globally to all interfaces on which NAT is enabled.

- Dynamic timeout – This age timer applies to all entries (static and dynamic) that do not use Port Address Translation. The default is 120 seconds.
- UDP timeout – This age timer applies to entries that use Port Address Translation based on UDP port numbers. The default is 120 seconds.
- TCP timeout – This age timer applies to entries that use Port Address Translation based on TCP port numbers. The default is 120 seconds.

NOTE: This timer applies only to TCP sessions that do not end “gracefully”, with a TCP FIN or TCP RST.

- TCP FIN/RST timeout – This age timer applies to TCP FIN (finish) and RST (reset) packets, which normally terminate TCP connections. The default is 120 seconds.

  NOTE: This timer is not related to the TCP timeout. The TCP timeout applies to packets to or from a host address that is mapped to an global IP address and a TCP port number (Port Address Translation feature). The TCP FIN/RST timeout applies to packets that terminate a TCP session, regardless of the host address or whether Port Address Translation is used.

- DNS timeout – This age timer applies to connections to a Domain Name Server (DNS). The default is 120 seconds.

To change the timeout for a dynamic entry type, use the following CLI method.

USING THE CLI

To change the age timeout for all entries that do not use Port Address Translation to 1800 seconds (one half hour), enter a command such as the following at the global CONFIG level of the CLI:

```
BigIron(config)# ip nat timeout 1800
```

Syntax: [no] ip nat translation timeout | udp-timeout | tcp-timeout | finrst-timeout | dns-timeout <secs>

Use one of the following parameters to specify the dynamic entry type:

- **timeout** – All entries that do not use Port Address Translation. The default is 120 seconds.
- **udp-timeout** – Dynamic entries that use Port Address Translation based on UDP port numbers. The default is 120 seconds.
- **tcp-timeout** – Dynamic entries that use Port Address Translation based on TCP port numbers. The default is 120 seconds.
- **finrst-timeout** – TCP FIN (finish) and RST (reset) packets, which normally terminate TCP connections. The default is 120 seconds.
- **dns-timeout** – Connections to a Domain Name Server (DNS). The default is 120 seconds.

The <secs> parameter specifies the number of seconds. For each entry type, you can enter a value from 1 – 3600.

Changing the Time a Session Table Entry Stays in the Delete Queue

Upon receiving a FIN from the client, the software puts the session in a delete queue and ages out the session table entry in eight seconds. To set the amount of time a session table entry stays in the delete queue , enter a command such as the following:

```
BigIron(config)#ip session tcp-msl 16
```

Syntax: [no] ip session tcp-msl <seconds>

The <seconds> parameter can be from 0 – 40 seconds. The default is 8 seconds.

To use the default, enter **ip session tcp-msl 0** or **no ip session tcp-msl 16**.

Displaying the Active NAT Translations

To display the currently active NAT translations, display the NAT translation table using the following CLI method.

NOTE: For information about the aging timer for NAT translation entries, see “Changing Translation Table Timeouts” on page 18-16.

USING THE CLI

To display the currently active NAT translations, enter the following command at any level of the CLI:

```
BigIron(config)# show ip nat translation
Pro Inside global      Inside local       Outside local      Outside global
--- 209.157.1.69     10.10.10.69       207.195.2.12      207.195.2.12
--- 209.157.1.72     10.10.10.2        207.195.4.69      207.195.4.69
```

Syntax: show ip nat translation

NOTE: This command does not display ICMP translations.

The **show ip nat translation** command shows the following information.

Table 18.1: CLI Display of Active NAT Translations

This Field...	Displays...
Pro	When Port Address Translation is enabled, this field indicates the protocol NAT is using to uniquely identify the host. NAT can map the same IP address to multiple hosts and use the protocol port to distinguish among the hosts. This field can have one of the following values: • tcp – In addition to this IP address, NAT is associating a TCP port with the host on the private network. • udp – In addition to this IP address, NAT is associating a UDP port with the host on the private network.
Inside global	The Internet address mapped to the private address listed in the Inside local field for inside NAT.
Inside local	The private address mapped to the Internet private address listed in the Inside global field for inside NAT.
Outside global	The destination of the traffic. If Port Address Translation is enabled, the TCP or UDP port also is shown.
Outside local	In the current release, the same as Outside global.

Displaying NAT Statistics

To display NAT statistics, use the following CLI method.

USING THE CLI

To display the NAT statistics, enter the following command at any level of the CLI:

```
BigIron(config)# show ip nat statistics

Total translations: 10 (0 static, 10 dynamic)
Hits: 10 Misses: 1
Expired translations: 1
Dynamic mappings:
 pool rtrpool: mask = 255.255.255.255
     start 192.168.2.79 end 192.168.2.79
     total addresses 1 overloaded
IP Fragments: saved 0, restored 0, timed out 0
Sess: Total 524288, Avail 524243, NAT 22

Inside global      Last Inside Local  xmit pkts  xmit bytes rx pkts    rx bytes  cnt
192.168.2.79       10.10.100.18       62         4012       42         4285      10
```

Syntax: show ip nat statistics

The **show ip nat statistics** command shows the following information.

Table 18.2: CLI Display of NAT Statistics

This Field...	Displays...
Total translations	The number of translations that are currently active. This number changes when translations are added or age out. To display the currently active translations, enter the **show ip nat translation** command.
Hits	The number of times NAT searched the translation table for a NAT entry and found the needed entry. (To optimize performance, NAT looks in the NAT table for an existing entry for a given translation before creating an entry for that translation.)
Misses	The number of times NAT did not find a needed entry in the translation table. When this occurs, NAT creates the needed entry and places it in the table.
Expired translations	The total number of dynamic translations that have aged of the translation table since the Foundry device was booted.

Table 18.2: CLI Display of NAT Statistics (Continued)

This Field...	Displays...
Dynamic mappings	Lists the dynamic translation parameters configured for the device. The following information is displayed: • pool – The name of the pool from which the address used for the translation was drawn. • mask – The subnet mask or prefix used for addressed in the pool. • start – The beginning (lowest) IP address in the pool. • end – The ending (highest) IP address in the pool. • total addresses – The total number of active address translations that are based on addresses in this pool. In addition, if the pool uses the Port Address Translation feature, the word “overloaded” appears at the end of this row.
IP Fragments	Lists statistics for fragmented packets: • saved – The number of out-of-sequence IP fragments saved. • restored – The number of saved out-of-sequence IP fragments that were successfully forwarded. • timed out – The number of saved out-of-sequence IP fragments that were dropped because the first IP fragment was never received.
Sess	Lists session statistics. NAT uses the session table for managing the translations. • Total – The total number of both used and available internal session resources. • Avail – The number of free internal session resources. • NAT – The number of internal session resources currently used by NAT. For information about the session table, see “Layer 4 Session Table” on page 12-7.
Inside global	A global IP address.
Last Inside Local	The last inside local IP address to use the global IP address.
xmit pkts	The number of packets send out for this NAT global IP address from the inside to the outside network.
xmit bytes	The number of bytes send out for this NAT global IP address from the inside to the outside network.
rx pkts	The number of packets received from the outside network to the inside network for this NAT global IP address.
rx bytes	The number of bytes received from the outside network to the inside network for this NAT global IP address.

Table 18.2: CLI Display of NAT Statistics (Continued)

This Field...	Displays...
cnt	The number of session resources in use for the translation. **Note**: If the value is 0, then translation is not taking place. Check your configuration. For example, make sure you have enabled both inside NAT (on the interface to the private addresses) and outside NAT (on the interface to the Internet).

Clearing Translation Table Entries

In addition to the aging mechanism, the software allows you to manually clear entries from the NAT table. The software provides the following clear options:

- Clear all entries (static and dynamic)
- Clear an entry for a specific NAT entry based on the private and global IP addresses
- Clear an entry for a specific NAT entry based on the IP addresses and the TCP or UDP port number. Use this option when you are trying to clear specific entries created using the Port Address Translation feature.

To clear entries, use the following CLI method.

USING THE CLI

To clear all dynamic entries from the NAT translation table, enter the following command at the Privileged EXEC level of the CLI:

```
BigIron# clear ip nat all
```

Syntax: clear ip nat all

To clear only the entries for a specific address entry, enter a command such as the following:

```
BigIron# clear ip nat inside 209.157.1.43 10.10.10.5
```

This command clears the inside NAT entry that maps private address 10.10.10.5 to Internet address 209.157.1.43. Here is the syntax for this form of the command.

Syntax: clear ip nat inside <global-ip> <private-ip>

If you use Port Address Translation, you can selectively clear entries based on the TCP or UDP port number assigned to an entry by the feature. For example, the following command clears one of the entries associated with Internet address 209.157.1.44 but does not clear other entries associated with the same address.

```
BigIron# clear ip nat inside 209.157.1.43 1081 10.10.10.5 80
```

The command above clears all inside NAT entries that match the specified global IP address, private IP address, and TCP or UDP ports.

Syntax: clear ip nat <protocol> inside <global-ip> <internet-tcp/udp-port> <private-ip> <private-tcp/udp-port>

The <protocol> parameter specifies the protocol type and can be **tcp** or **udp**.

NAT Debug Commands

To configure the device to display diagnostic information for NAT, enter a **debug ip nat** command.

Syntax: [no] debug ip nat icmp | tcp | udp <ip-addr>

Syntax: [no] debug ip nat transdata

The <ip-addr> parameter specifies an IP address. The address applies to packets with the address as the source or the destination. Specify 0.0.0.0 to enable the diagnostic mode for all addresses.

The following examples show sample output from **debug ip nat** commands. The first three examples show the output from the diagnostic mode for ICMP NAT, TCP NAT, and UDP NAT. The fourth command shows the output for the diagnostic mode for NAT translation requests.

```
BigIron# debug ip nat icmp 192.168.3.11
NAT: ICMP debugging is on
NAT: icmp src 10.10.100.18 => trans 192.168.2.78 dst 192.168.3.11
NAT: ICMP src 10.10.100.18 => trans 192.168.2.78 dst 192.168.3.11
NAT: 192.168.2.78 192.168.3.11 ID 60950 len 60 txfid 13 icmp (8/0/512/13824)
NAT: ICMP dest 192.168.2.78 => trans 192.168.3.11 dst 10.10.100.18
NAT: 192.168.3.11 10.10.100.18 ID 5571 len 60 txfid 15 icmp (0/0/512/13824)
NAT: icmp src 10.10.100.18 => trans 192.168.2.78 dst 192.168.3.11
NAT: ICMP src 10.10.100.18 => trans 192.168.2.78 dst 192.168.3.11
NAT: 192.168.2.78 192.168.3.11 ID 61206 len 60 txfid 13 icmp (8/0/512/14080)
NAT: ICMP dest 192.168.2.78 => trans 192.168.3.11 dst 10.10.100.18
NAT: 192.168.3.11 10.10.100.18 ID 5572 len 60 txfid 15 icmp (0/0/512/14080)
NAT: icmp src 10.10.100.18 => trans 192.168.2.78 dst 192.168.3.11
NAT: ICMP src 10.10.100.18 => trans 192.168.2.78 dst 192.168.3.11
NAT: 192.168.2.78 192.168.3.11 ID 61462 len 60 txfid 13 icmp (8/0/512/14336)
NAT: ICMP dest 192.168.2.78 => trans 192.168.3.11 dst 10.10.100.18
NAT: 192.168.3.11 10.10.100.18 ID 5573 len 60 txfid 15 icmp (0/0/512/14336)
```

```
BigIron# debug ip nat tcp 192.168.3.11
NAT: TCP debugging is on
NAT: tcp src 10.10.100.18:1144 => trans 192.168.2.78:8012 dst 192.168.3.11:53
NAT: tcp data src 10.10.100.18:1144 => trans 192.168.2.78:8012 dst 192.168.3.11:53
NAT: 192.168.2.78:8012 192.168.3.11:53 flags S ID 64534 len 44 txfid 13
NAT: tcp data dest 192.168.2.78:8012 => trans 192.168.3.11:53 dst
10.10.100.18:1144
NAT: 192.168.3.11:53 10.10.100.18:1144 flags S A ID 64921 len 44 txfid 15
NAT: tcp data src 10.10.100.18:1144 => trans 192.168.2.78:8012 dst 192.168.3.11:53
NAT: 192.168.2.78:8012 192.168.3.11:53 flags A ID 64790 len 40 txfid 13
NAT: tcp data src 10.10.100.18:1144 => trans 192.168.2.78:8012 dst 192.168.3.11:53
NAT: 192.168.2.78:8012 192.168.3.11:53 flags A ID 65046 len 78 txfid 13
NAT: tcp data dest 192.168.2.78:8012 => trans 192.168.3.11:53 dst
10.10.100.18:1144
NAT: 192.168.3.11:53 10.10.100.18:1144 flags A ID 64922 len 147 txfid 15
NAT: tcp data src 10.10.100.18:1144 => trans 192.168.2.78:8012 dst 192.168.3.11:53
NAT: 192.168.2.78:8012 192.168.3.11:53 flags A ID 65302 len 40 txfid 13
NAT: tcp data src 10.10.100.18:1144 => trans 192.168.2.78:8012 dst 192.168.3.11:53
NAT: 192.168.2.78:8012 192.168.3.11:53 flags FA ID 23 len 40 txfid 13
NAT: tcp data dest 192.168.2.78:8012 => trans 192.168.3.11:53 dst
10.10.100.18:1144
NAT: 192.168.3.11:53 10.10.100.18:1144 flags A ID 64923 len 40 txfid 15
NAT: tcp data dest 192.168.2.78:8012 => trans 192.168.3.11:53 dst
10.10.100.18:1144
NAT: 192.168.3.11:53 10.10.100.18:1144 flags FA ID 64924 len 40 txfid 15
NAT: tcp data src 10.10.100.18:1144 => trans 192.168.2.78:8012 dst 192.168.3.11:53
NAT: 192.168.2.78:8012 192.168.3.11:53 flags A ID 279 len 40 txfid 13
```



```
BigIron# debug ip nat udp 192.168.3.11
NAT: udp src 10.10.100.18:1140 => trans 192.168.2.78:8008 dst 192.168.3.11:53
NAT: udp data src 10.10.100.18:1140 => trans 192.168.2.78:8008 dst 192.168.3.11:53
NAT: 192.168.2.78:8008 192.168.3.11:53 ID 54806 len 63 txfid 13
NAT: udp src 10.10.100.18:1141 => trans 192.168.2.78:8009 dst 192.168.3.11:53
NAT: udp data src 10.10.100.18:1141 => trans 192.168.2.78:8009 dst 192.168.3.11:53
NAT: 192.168.2.78:8009 192.168.3.11:53 ID 55062 len 63 txfid 13
NAT: udp data dest 192.168.2.78:8008 => trans 192.168.3.11:53 dst
10.10.100.18:1140
NAT: 192.168.3.11:53 10.10.100.18:1140 ID 56965 len 246 txfid 15
NAT: udp data dest 192.168.2.78:8009 => trans 192.168.3.11:53 dst
10.10.100.18:1141
NAT: 192.168.3.11:53 10.10.100.18:1141 ID 56966 len 246 txfid 15
```

```
BigIron# debug ip nat transdata
NAT: icmp src 10.10.100.18:2048 => trans 192.168.2.79 dst 204.71.202.127
NAT: udp  src 10.10.100.18:1561 => trans 192.168.2.79:65286 dst 192.168.3.11:53
NAT: tcp  src 10.10.100.18:1473 => trans 192.168.2.78:8016 dst 192.168.2.158:53
```


To disable the NAT diagnostic mode, enter a command such as the following:

```
BigIron# no debug ip nat tcp
```

This command disables the diagnostic mode for NAT performed on TCP packets. NAT diagnostics for other types of packets remain enabled.

You also can use the following syntax to disable the diagnostic mode for NAT:

Syntax: undebug ip nat icmp | tcp | udp | transdata

Chapter 19
Configuring VRRP and VRRPE

This chapter describes how to configure Foundry Layer 3 Switches to configure 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: 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.

Foundry Standby Router Protocol (FSRP), a Foundry router redundancy protocol available before VRRP or VRRPE, is described in “Configuring FSRP” on page 21-1.

For a summary of how these two router redundancy protocols differ, see “Comparison of VRRP, VRRPE, and FSRP” on page 19-8.

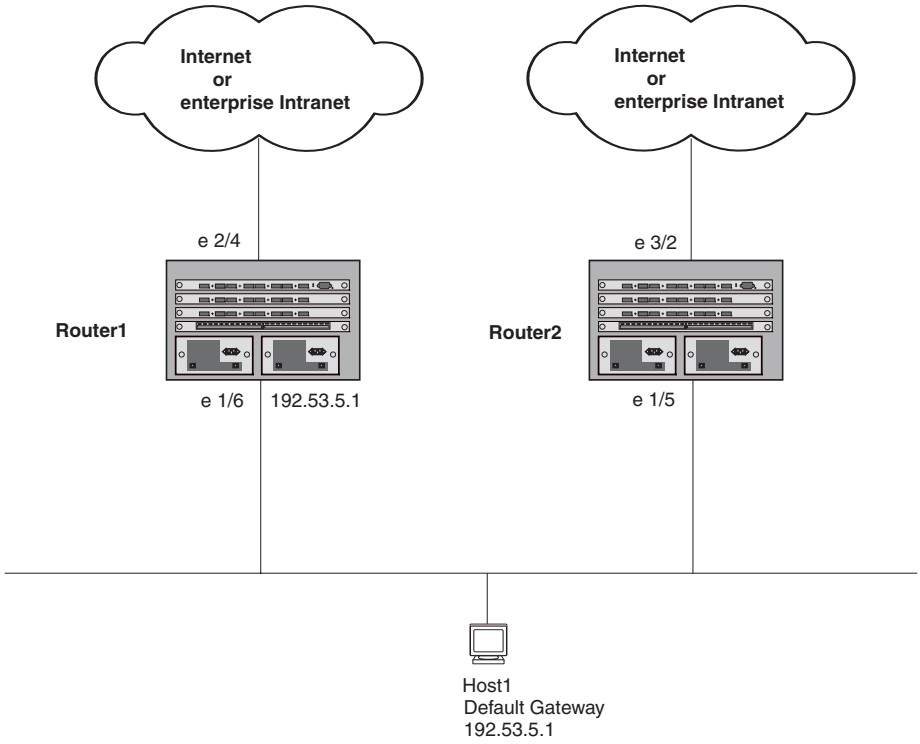
Overview

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

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 19.1.

Figure 19.1 Router1 is Host1's default gateway but is a single point of failure

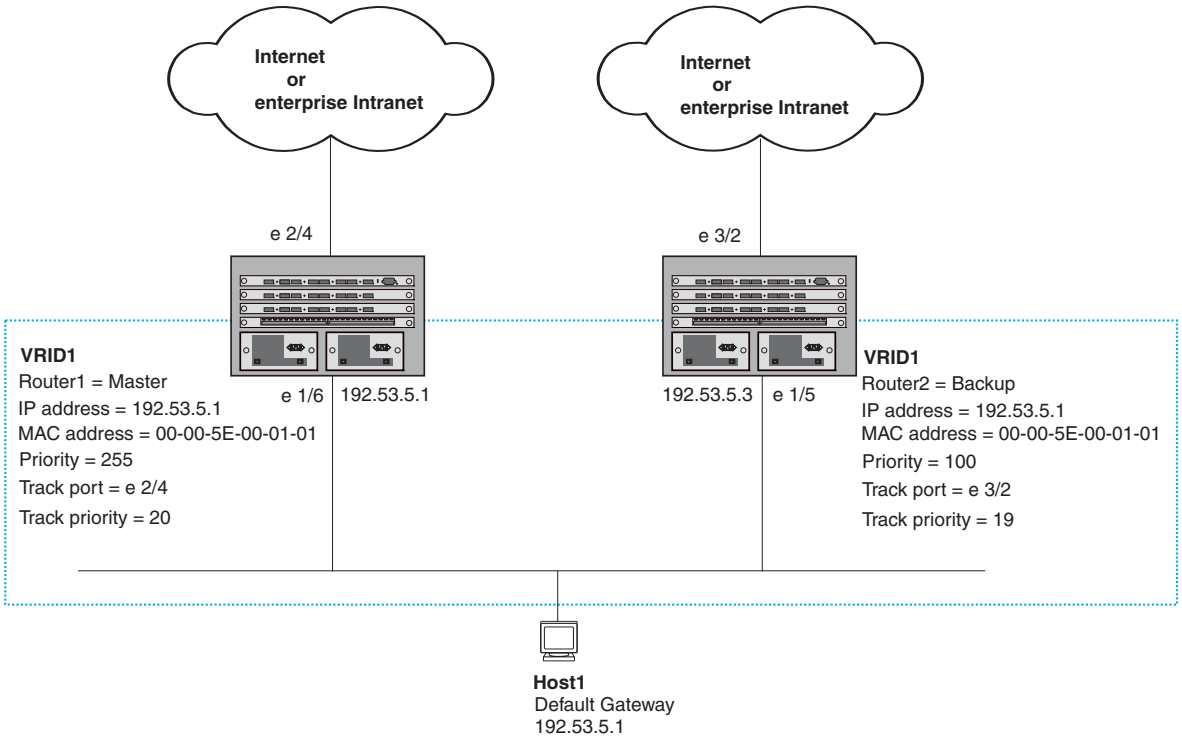


As shown in this example, Host1 uses 192.53.5.1 on Router1 as the host's default gateway out of the subnet. If this interface goes down, Host1 is cut off from the rest of the network. Router1 is thus a single point of failure for Host1's access to other networks.

If Router1 fails, you could configure Host1 to use Router2. 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 Router1 and Router2 to provide a redundant path for the host(s).

Figure 19.2 shows the same example network shown in Figure 19.1, but with a VRRP virtual router configured on Router1 and Router2.

Figure 19.2 Router1 and Router2 are configured as a VRRP virtual router to provide redundant network access for Host1



The dashed box in Figure 19.2 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 Router2 as the Owner and Router1 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 19.2, Router1 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

Unlike Foundry Standby Router Protocol (FSRP), 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 19.2, the virtual router with VRID1 is associated with real IP address 192.53.5.1, which is configured on interface e1/6 on Router1. 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. See the "Using Packet Over SONET Modules" chapter in the *Foundry Switch and Router Installation and Basic Configuration Guide*.

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 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 address(es) you plan to associate with the VRID or a Backup. If you indicate that the router is the Owner of the IP address(es), 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 address(es) 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 19-5.

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 19-18.

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 19.2 on page 19-3, interface e1/6 on Router1 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, Router1 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 Router1's VRRP priority to the value of the track priority. In the configuration shown in Figure 19.2 on page 19-3, Router1'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 19.2 on page 19-3, the track priority results in Router1's VRRP priority becoming lower than Router2's VRRP priority. As a result, when Router2 learns that it now has a higher priority than Router1, Router2 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 address(es) 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 address(es) 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.

VRRP Support on FES Devices

FES software release 03.4.00 supports VRRP in the base Layer 3 code. Previous releases support VRRP in the full Layer 3 code only. VRRP support in the base Layer 3 code is the same as in the full Layer 3 code.

NOTE: NOTE: VRRP-E is supported in the full Layer 3 code only. It is not supported in the base Layer 3 code.

Overview of VRRPE

VRRPE is similar to VRRP, but differs in the following respects:

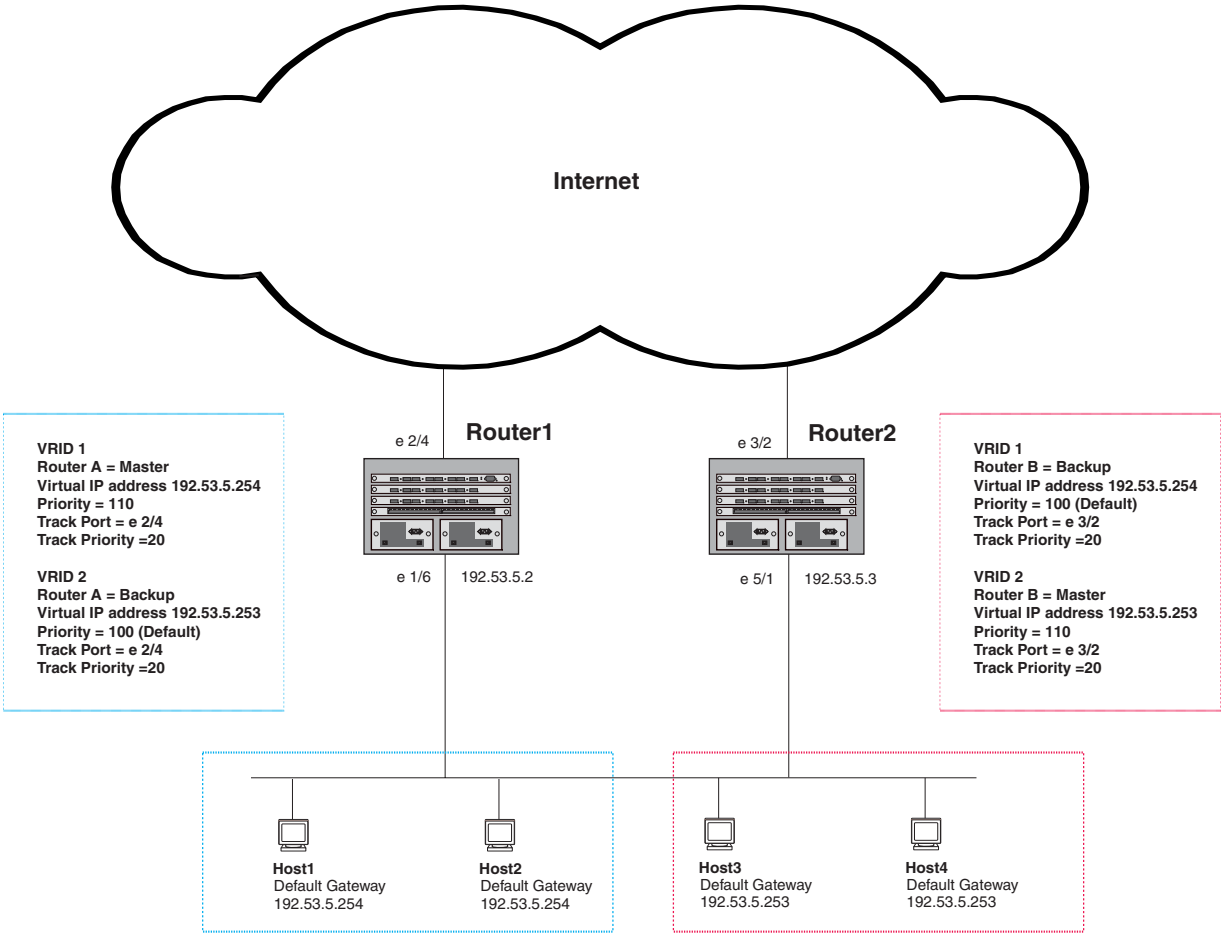
- Owners and Backups
  - VRRP has an Owner and one or more Backups for each VRID. The Owner is the router on which the VRID's IP address is also configured as a real address. All the other routers supporting the VRID are Backups.
  - VRRPE does not use Owners. All routers are Backups for a given VRID. 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 19.3 shows an example of a VRRPE configuration.

Figure 19.3 Router1 and Router2 are configured to provide dual redundant network access for the host



In this example, RouterA and RouterB 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 RouterA and the rest to go through RouterB.

RouterA is the master for VRID 1 (backup priority = 110) and RouterB is the backup for VRID 1 (backup priority = 100). RouterA and RouterB both track the uplinks to the Internet. If an uplink failure occurs on RouterA, its backup priority is decremented by 20 (track priority = 20), so that all traffic destined to the Internet is sent through RouterB instead.

Similarly, RouterB is the master for VRID 2 (backup priority = 110) and RouterA is the backup for VRID 2 (backup priority = 100). RouterA and RouterB are both tracking the uplinks to the Internet. If an uplink failure occurs on RouterB, its backup priority is decremented by 20 (track priority = 20), so that all traffic destined to the internet is sent through RouterA instead.

Comparison of VRRP, VRRPE, and FSRP

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 19-5.
- 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. In fact, VRRPE combines the benefits of Foundry's VRRP and FSRP (see "FSRP"). VRRPE is unlike VRRP and is like FSRP 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.

FSRP

The Foundry ***Standby Router Protocol*** (FSRP) is another Foundry router redundancy protocol that provides many of the same features as Foundry's implementation of VRRP and VRRPE. However, FSRP does not provide authentication, which VRRP and VRRPE do. In addition, FSRP allows only one backup router.

FSRP is available only on 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").
- FSRP – FSRP sends management traffic to a user-configured unicast address.

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.
- FSRP – The virtual router IP address is a user-configured virtual IP address.

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.
- FSRP – You can configure one Primary Router and one Backup Router. There is no Owner. You must define the virtual IP address (the one you are backing up) on both the Primary Router and the Backup Router.

NOTE: If your Foundry routers already are using FSRP and you do not need redundancy with devices that cannot use FSRP, you do not need to reconfigure your routers to use VRRP or VRRPE.

Foundry Networks recommends that you do not use more than one redundancy protocol (VRRP, VRRPE, or FSRP) on the same device.

VRRP and VRRPE Parameters

Table 19.1 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 19.1: 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.	19-12 19-13
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	19-12 19-13
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	19-3 19-12 19-13

Table 19.1: VRRP and VRRPE Parameters (Continued)

Parameter	Description	Default	See page...
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	19-4 19-12 19-13
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	19-3
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	19-5 19-14
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.	19-15

Table 19.1: VRRP and VRRPE 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 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	19-15
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	19-16
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	19-4 19-17
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	19-4 19-17
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	19-4 19-17
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	19-5 19-18

Table 19.1: VRRP and VRRPE Parameters (Continued)

Parameter	Description	Default	See page...
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	19-5 19-18
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	19-18

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)# 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 address(es) associated with the VRID must already be configured on the router that will be the Owner router.
- An IP address(es) 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 Backup(s).

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)# 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 address(es) 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:

```
BigIron(config-vrrp-router)# no router vrrp
router vrrp mode now disabled. All vrrp 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 vrrp**), 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 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
- Slow start

For VRRP, you can set some of these parameters using the VRRP configuration panel of the Web management interface, shown in "" on page 19-47. For information about the fields, see the parameter descriptions in the following sections. To access this panel, select VRRP from the System configuration sheet, then click Modify next to the VRRP entry you want to edit.

NOTE: You cannot set VRRPE parameters using the Web management interface.

See "VRRP and VRRPE Parameters" on page 19-9 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.

USING THE CLI

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 19-19.

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.

USING THE CLI

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
```

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:

```
BigIron(config)# inter e 1/1
BigIron(config-if-1/1)# ip vrrp-extended vrid 1
BigIron(config-if-1/1-vrid-1)# backup priority 50 track-priority 10
```

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 Up 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.

USING THE CLI

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.

USING THE CLI

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.

USING THE CLI

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.

USING THE CLI

To enable a Backup to send Hello messages to the Master, enter commands such as the following:

```
BigIron(config)# router vrrp
BigIron(config)# inter e 1/6
BigIron(config-if-1/6)# ip vrrp vrid 1
BigIron(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:

```
BigIron(config)# router vrrp
BigIron(config)# inter e 1/6
BigIron(config-if-1/6)# ip vrrp vrid 1
BigIron(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 19-5.

USING THE CLI

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 <portnum> | pos <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 19-18.

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.

USING THE CLI

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.

VRRPE Slow Start Timer

In a VRRPE 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. In this release, you can configure the VRRPE 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.

To set the VRRPE slow start timer to 30 seconds, enter the following command:

```
BigIron(config)# vrrp-e slow-start 30
```

Syntax: [no] vrrp-e slow-start <seconds>

When the VRRPE 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 VRRPE slow start timer elapses (in this example, 30 seconds) before the Master takes over from the Backup.

The VRRPE slow start timer is effective only if another VRRPE Master (Standby) is detected. It is not effective during the initial bootup.

NOTE: The VRRPE slow start timer applies only to VRRPE 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.

USING THE CLI

To change the Master's priority, enter commands such as the following:

```
BigIron(config)# ip int eth 1/6
BigIron(config-if-1/6)# ip vrrp vrid 1
BigIron(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:

```
BigIron(config-if-1/6-vrid-1)# 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:

```
BigIron(config-if-1/6-vrid-1)# no owner priority 99
```

You cannot set the priority to 255 using the **owner priority** command.

USING THE WEB MANAGEMENT INTERFACE

You cannot change the Master router's priority using the Web management interface.

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

NOTE: You cannot display VRRPE information using the Web management interface.

Displaying Summary Information

To display summary VRRP or VRRPE information, use the following CLI method.

USING THE CLI

To display summary information for a Layer 3 Switch, enter the following command at any level of the CLI:

```
BigIron(config-if-e1000-1/6-vrid-1)# 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
```

This example is for VRRP. Here is an example for VRRPE:

```
BigIron(config-if-e1000-1/6-vrid-1)# 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 <portnum> | ve <num> | stat

Syntax: show ip vrrp-extended brief | ethernet <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 19-22.

The **ethernet** <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 19-29.

This display shows the following information.

Table 19.2: 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.

Table 19.2: CLI Display of VRRP or VRRPE Summary 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: • 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	The IP address of the router interface that is currently the Master for the VRID.
Backup addr	The IP addresses of the router interfaces that are currently Backups for the VRID.
VIP	The virtual IP address that is being backed up by the VRID.

USING THE WEB MANAGEMENT INTERFACE

You cannot display the summary view using the Web management interface. Use the Web management procedure in "Displaying Detailed Information".

Displaying Detailed Information

To display detailed VRRP or VRRPE information, use either of the following methods.

USING THE CLI

To display detailed information, enter the following command at any level of the CLI:

```
BigIron(config)# 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 1 sec
  advertise backup: disabled
  track-port 2/4
```

This example is for a VRRP Owner. Here is an example for a VRRP Backup.

```
BigIron(config)# 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 1 sec
  dead-interval 3.600 sec
  current dead-interval 3.600 sec
  preempt-mode true
  advertise backup: enabled
  backup router 192.53.5.3 expires in 00:00:03
  next hello sent in 00:00:02
  track-port 3/2
```

Here is an example for a VRRPE Backup.

```
BigIron(config)# 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 1 sec
  dead-interval 3.600 sec
  current dead-interval 3.600 sec
  preempt-mode true
  virtual ip address 192.53.5.254
  advertise backup: enabled
  master router 192.53.5.2 expires in 00:00:03
  track-port 2/4
```

Syntax: show ip vrrp brief | ethernet <portnum> | ve <num> | stat

Syntax: show ip vrrp-extended brief | ethernet <portnum> | ve <num> | stat

The **brief** parameter displays summary information. See “Displaying Summary Information” on page 19-20.

The **ethernet** <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 19-29.

This display shows the following information.

Table 19.3: 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.
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 backup up by the VRID. **Note**: This field applies only to VRRP. All Layer 3 Switches configured for VRRPE are Backups.

Table 19.3: CLI Display of VRRP or VRRPE Detailed Information (Continued)

This Field...	Displays...
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 19-5.
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. **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. **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 19-4.

Table 19.3: CLI Display of VRRP or VRRPE Detailed Information (Continued)

This Field...	Displays...
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).
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.

USING THE WEB MANAGEMENT INTERFACE

NOTE: This procedure applies only to VRRP. You cannot display VRRPE information using the Web management interface.

1. Log on to the device using a valid user name and password for read-only or read-write access. The System configuration dialog is displayed.
2. Click on the plus sign next to Monitor in the tree view to display the monitoring options.
3. Click on the plus sign next to VRRP in the tree view to expand the list of VRRP option links.
4. Click on the Interface link to display the virtual router table.
5. Click on the Virtual Router link to display the virtual router table.

NOTE: If a parameter is not defined or does not apply to this type of entry, the field is blank. For example, if the entry is for a VRRP Owner, the Backup Priority field does not apply and is blank.

This display shows the following information.

Table 19.4: Web Display of VRRP Detailed Information

This Field...	Displays...
Interface table	
Port	The interface number. All the device's interfaces are listed.
Authentication Type	The authentication type enabled on the interface.
Simple Text Password	If the authentication type is simple password, this field lists the password.
Virtual Router table	
Port	The interface number. All the device's interfaces are listed.
ID	The VRID configured on this interface. If multiple VRIDs are configured on the interface, information for each VRID is listed separately.
Hello Intv	The number of seconds between Hello messages from the Master to the Backups for a given VRID.
Activate	Indicates whether this VRID is activated. After configuring the VRID, you must activate it. The VRID is disabled by default.
IP List	The IP addresses that this VRID is backing up.
Mode	Indicates whether the Layer 3 Switch is the Owner or a Backup for the VRID. **Note**: The mode applies only to VRRP. All Layer 3 Switches configured for VRRPE are Backups.
Backup – 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.
Backup – Dead Intv	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.
Backup – Preempt	The state of the Backup preempt mode. The 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.
Track – Priority	A VRRP priority value assigned to the tracked port(s). If a tracked port's link goes down, the VRID port's priority is reduced by the amount of the tracked port's priority.

Table 19.4: Web Display of VRRP Detailed Information (Continued)

This Field...	Displays...
Track – Vif List	The virtual interfaces that the VRID's interface is tracking. If the link for a tracked interface goes down, the VRRP priority of the VRID interface is changed, causing the devices to renegotiate for Master.
Track – Port List	The physical ports that the VRID's interface is tracking. If the link for a tracked port goes down, the VRRP priority of the VRID interface is changed, causing the devices to renegotiate for Master.

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:

```
BigIron(config)# 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 19.5: Output from the show ip vrrp vrid command

This Field...	Displays...
VRID	The specified VRID.
Interface	The interface on which VRRP is configured.

Table 19.5: Output from the show ip vrrp vrid command (Continued)

This Field...	Displays...
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 configured number of seconds 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 VRRP or VRRPE statistics, use either of the following methods.

USING THE CLI

To display statistics on most Foundry devices, enter a command such as the following at any level of the CLI:

```
BigIron(config-if-e1000-1/5-vrid-1)# 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 <portnum> | ve <num> | statistic

Syntax: show ip vrrp-extended brief | ethernet <portnum> | ve <num> | stat

The **brief** parameter displays summary information. See “Displaying Summary Information” on page 19-20.

The **ethernet** <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 19-22.

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 19-22.

The **statistic** parameter displays statistics. This parameter is required for displaying the statistics.

This display shows the following information.

Table 19.6: 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.
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.

Table 19.6: CLI Display of VRRP or VRRPE Statistics (Continued)

This Field...	Displays...
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.

To display VRRP statistics on BigIron MG8 software release 02.0.02 and later and NetIron 40G devices software release 02.2.04 and later, enter the following command:

```
BigIron MG8#show ip vrrp statistics

Global VRRP statistics
----------------------
- received vrrp packets with checksum errors = 0
- received vrrp packets with invalid version number = 0
- received vrrp packets with unknown or inactive vrid = 0

Interface 1/1
---------------
VRID 1
- number of transitions to backup state = 2
- number of transitions to master state = 1
- total number of vrrp packets received = 129
  . received backup advertisements = 0
  . received packets with zero priority = 1
  . received packets with invalid type = 0
  . received packets with invalid authentication type = 0
  . received packets with authentication type mismatch = 0
  . received packets with authentication failures = 0
  . received packets dropped by owner = 0
  . received packets with ip ttl errors = 0
  . received packets with ip address mismatch = 0
```



```
  . received packets with advertisement interval mismatch = 0
  . received packets with invalid length = 0
- total number of vrrp packets sent = 2018
  . sent backup advertisements = 0
  . sent packets with zero priority = 0
- received arp packets dropped = 0
- received proxy arp packets dropped = 0
- received ip packets dropped = 0
```

Syntax: show ip vrrp brief | ethernet <portnum> | ve <num> | stat

The **brief** parameter displays the summary information. If you do not use this parameter, detailed information is displayed instead.

The **ethernet** <portnum> parameter specifies an Ethernet port. If you use this parameter, the command displays VRRP information only for the specified port.

The **ve** <num> parameter specifies a virtual interface. If you use this parameter, the command displays VRRP information only for the specified virtual interface.

The **statistics** parameter displays statistics.

USING THE WEB MANAGEMENT INTERFACE

NOTE: This procedure applies only to VRRP. You cannot display VRRPE information using the Web management interface.

1. Log on to the device using a valid user name and password for read-only or read-write access. The System configuration dialog is displayed.
2. Click on the plus sign next to Monitor in the tree view to display the monitoring options.
3. Click on the plus sign next to VRRP in the tree view to expand the list of VRRP option links.
4. Click on the Virtual Router link to display the virtual router table or the Interface link to display the VRRP Interface table. The VRRP Interface table shows a row for each interface on the Layer 3 Switch.

NOTE: If a parameter is not defined or does not apply to this type of entry, the field is blank. For example, if the entry is for a VRRP Owner, the Backup Priority field does not apply and is blank.

NOTE: It is possible for the statistics display for a Backup to show “Master” in the state field even when you have not yet configured another VRRP or VRRPE router. When you activate a Backup, if the Backup’s Dead interval expires before the Backup hears from another VRRP or VRRPE router, the Backup becomes the Master.

This display shows the following information.

Table 19.7: Web Display of VRRP Statistics

This Field...	Displays...
Virtual Router panel	
Port	The interface on which VRRP is configured. If VRRP is configured on more than one interface, the display lists the statistics separately for each interface.
Header Error	The number of VRRP packets received by the interface that had a header error.
Authen Type Error	The number of VRRP packets received by the interface that had an authentication error.

Table 19.7: Web Display of VRRP Statistics (Continued)

This Field...	Displays...
Authen Password Mismatch Error	The number of VRRP packets received by the interface that had a password value that does not match the password used by the interface for authentication.
Virtual Router ID Error	The number of VRRP packets received by the interface that contained a VRID that is not configured on this interface.
Interface Statistics panel	
Port	The interface on which VRRP is configured. If VRRP is configured on more than one interface, the display lists the statistics separately for each interface.
ID	The VRID.
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.
Receive Pkts Drop – ARP	The number of ARP packets addressed to the VRID that were dropped.
Receive Pkts Drop – IP	The number of IP packets addressed to the VRID that were dropped.
Receive Mismatch – Port	The number of packets received that did not match the configuration for the receiving interface.
Receive Mismatch – Num of IP	The number of packets received that did not match the configured IP addresses.
Receive Mismatch – IP	The number of packets received that did not match the configured Hello interval.
Receive Mismatch – Hello	The current Master has resigned.
Rcv Priority Zero from Master	The number of packets received that did not match the configuration for the receiving interface.
Rcv Higher Priority	The number of VRRP 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.
Transmit Count – Master	The number of times this Layer 3 Switch has changed from the backup state to the master state for the VRID.
Transmit Count – Backup	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.

USING THE CLI

To clear VRRP or VRRPE statistics, enter the following command at the Privileged EXEC level or any configuration level of the CLI:

```
Router1(config)# clear ip vrrp-stat
```

Syntax: clear ip vrrp-stat

USING THE WEB MANAGEMENT INTERFACE

NOTE: This procedure applies only to VRRP. You cannot display VRRPE information using the Web management interface.

1. Log on to the device using a valid user name and password for read-write access. The System configuration dialog is displayed.
2. Click on the plus sign next to Command in the tree view to expand the list of command options.
3. Click on the Clear link to display the Clear panel.
4. Select VRRP.
5. Click the Apply button to implement the change.

Displaying CPU Utilization Statistics

You can display CPU utilization statistics for VRRP and other IP protocols.

USING THE CLI

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:

```
BigIron# 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:

```
BigIron# 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:

```
BigIron# 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.

USING THE WEB MANAGEMENT INTERFACE

You cannot display this information using the Web management interface.

Configuration Examples

The following sections contain the CLI commands and Web management options for implementing the VRRP and VRRPE configurations shown in Figure 19.2 on page 19-3 and Figure 19.3 on page 19-7.

NOTE: The Web management example applies only to VRRP. You cannot configure VRRPE using the Web management interface.

VRRP Example

To implement the VRRP configuration shown in Figure 19.2 on page 19-3, use either of the following methods.

USING THE CLI

Configuring Router1 Using the CLI

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 Using the CLI

To configure Router2 in Figure 19.2 on page 19-3 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 19-5.

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 <portnum> | pos <portnum> | ve <num>

Syntax: ip-address <ip-addr>

Syntax: activate

USING THE WEB MANAGEMENT INTERFACE

Use the following procedures to create a virtual router using the Web management interface.

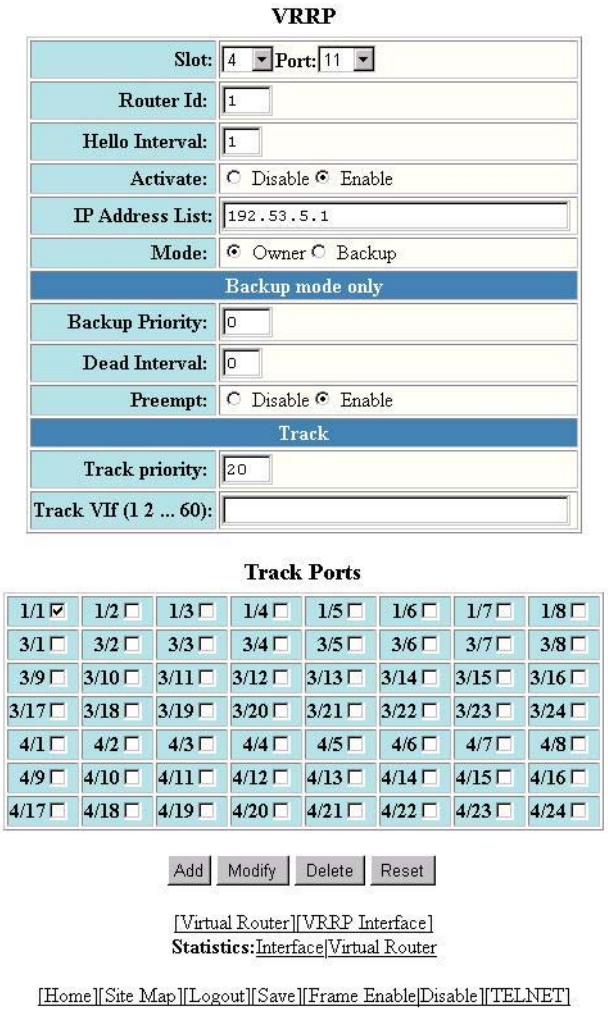
NOTE: Some of the data entry fields contain zeros. When you save a VRRP definition, the software uses the default values for the parameters instead of zeros. The Web management interface shows zeros instead of the defaults because the defaults differ depending on whether you are creating an Owner or a Backup. The software does not know which type of VRID entry you are creating until you select Add to add the entry.

Configuring Router1 Using the Web Management Interface

To configure VRRP Router1 in Figure 19.2 on page 19-3 after you enable VRRP:

1. Log on to the device using a valid user name and password for read-write access. The System configuration dialog is displayed.
2. Click on the plus sign next to Configure in the tree view to expand the list of configuration options.
3. Click on the plus sign next to VRRP in the tree view to expand the list of VRRP option links.
4. Click on the Virtual Router link.
   - If the device does not have a VRRP virtual router configured, the VRRP configuration panel is displayed, as shown in the following example.
   - If a VRRP virtual router is already configured and you are adding a new one, click on the Add Virtual Router link to display the VRRP configuration panel, as shown in the following example.
   - If you are modifying an existing VRRP virtual router, click on the Modify button to the right of the row describing the VRRP virtual router to display the VRRP configuration panel, as shown in the following

example.



5. Select the interface from the pulldown list on the Port field. In this example, select 1/6.
6. Enter the VRID in the Router ID field the Router ID field. In this example, use the default value, 1.
7. Enter the Hello interval or leave the field unchanged to use the default. The software fills in the default after you select Add. In this example, leave the field unchanged.
8. Select Enable to activate the VRRP entry after you select Add.
9. Enter the interface's IP address in the IP Address List field. In this example, enter 192.53.5.1.
10. Select the mode (Owner or Backup). Select Owner in this example.
11. Enter the track priority or leave the field blank to use the default. In this example, enter 20.
12. Enter or select the track interface or port:
    - If you want to use a virtual interface as a track port, enter the virtual interface name.
    - If you want to use a physical interface as a track port, select the port. In this example, select 2/4.
13. Click the Add button to apply the changes to the device's running-config.

14. Select the Save link at the bottom of the dialog, then select Yes when prompted to save the configuration change to the startup-config file on the device's flash memory.

Configuring Router2 Using the Web Management Interface

To configure VRRP Router2 in Figure 19.2 on page 19-3 after you enable VRRP:

1. Log on to the device using a valid user name and password for read-write access. The System configuration dialog is displayed.
2. Click on the plus sign next to Configure in the tree view to expand the list of configuration options.
3. Click on the plus sign next to VRRP in the tree view to expand the list of VRRP option links.
4. Click on the Virtual Router link.
   - If the device does not have a VRRP virtual router configured, the VRRP configuration panel is displayed.
   - If a VRRP virtual router is already configured and you are adding a new one, click on the Add Virtual Router link to display the VRRP configuration panel.
   - If you are modifying an existing VRRP virtual router, click on the Modify button to the right of the row describing the VRRP virtual router to display the VRRP configuration panel.
5. Select the interface from the pulldown list on the Port field. In this example, select 1/5.
6. Enter the VRID in the Router ID field the Router ID field. In this example, use the default value 1.
7. Enter the Hello interval or leave the field as is to use the default. The software fills in the default after you select Add. In this example, leave the field unchanged.
8. Select Enable to activate the VRRP entry after you select Add.
9. Enter the interface's IP address in the IP Address List field. In this example, enter 192.53.5.1. By entering the same IP address as the one associated with this VRID on the Owner, you configure 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.

10. Select the mode (Owner or Backup). Select Backup in this example.
11. Enter the backup priority or leave the value unchanged. In this example, enter 100.

    NOTE: This is the default for Backups. You also can leave the field unchanged, and the software will automatically assign 100 as the priority when you select Add.

12. Enter the Dead interval or leave the field unchanged to use the default value.
13. Enable preempt mode if desired. In this example, leave preempt mode disabled.
14. Enter the track priority or leave the field blank to use the default. In this example, enter 19.
15. Enter or select the track interface or port:
    - If you want to use a virtual interface as a track port, enter the virtual interface name.
    - If you want to use a physical interface as a track port, select the port. In this example, select 3/2.
16. Click the Add button to apply the changes to the device's running-config.
17. Select the Save link at the bottom of the dialog, then select Yes when prompted to save the configuration change to the startup-config file on the device's flash memory.

VRRPE Example

To implement the VRRPE configuration shown in Figure 19.3 on page 19-7, use the following CLI method.

Configuring Router1 Using the CLI

To configure VRRP Router1 in Figure 19.3 on page 19-7, 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 Using the CLI

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 19-5.

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 <portnum> | pos <portnum> | ve <num>

Syntax: ip-address <ip-addr>

Syntax: activate

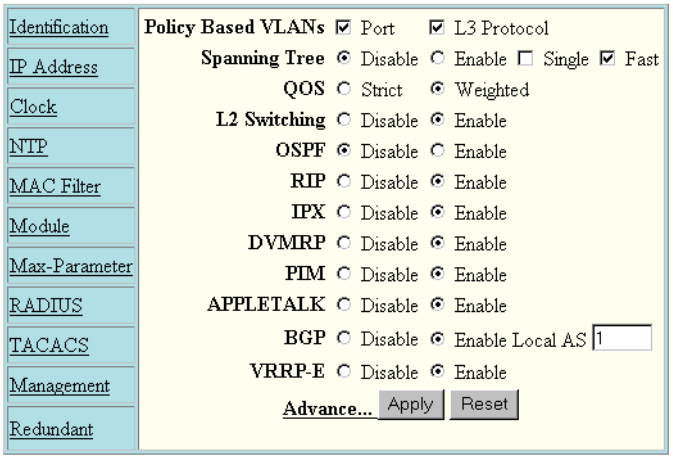
USING THE WEB MANAGEMENT INTERFACE

Enabling VRRPE

To enable VRRPE using the Web management interface, do the following:

1. Log on to the device using a valid user name and password for read-write access. The System configuration panel is displayed.
2. If VRRPE is not displayed on the System configuration panel, click Disable next to the virtual router that is currently enabled, then click Apply.

NOTE: Only one of the virtual router protocols (VRRP, VRRPE, or VSRP) can be enabled. Also, configuration data will be lost whenever you disable any of the virtual router protocol.



3. Click on Enable next to VRRPE to enable it.
4. Click on the Apply button to apply your changes.

5. 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.

After enabling VRRPE, options for other virtual routers (such as VRRP) are removed from the System configuration panel.

Configuring VRRPE Parameters

Once VRRPE is enabled, default values are applied to the ports on the Layer 3 Switch. You can change the values for individual ports by doing the following:

1. Log on to the device using a valid user name and password for read-write access. The System configuration panel is displayed.
2. Click on the plus sign next to Configure in the tree view to expand the list of configuration options.
3. Click on the plus sign next to Virtual Redundant Router to expand the list of Virtual Router options.
4. Click on the plus sign next to VRRP-Extended to display its list of options.
5. Click on the Virtual Router link to display the VRRPE parameters.
   - If Virtual Routers have been configured for the device, you see a list as in Figure 19.4. Click on Modify to make changes to the VRRPE configuration, or click on the Add virtual router link to add another virtual router.

Figure 19.4 List of Configured VRRPE Routers

VRRP-E Virtual Router

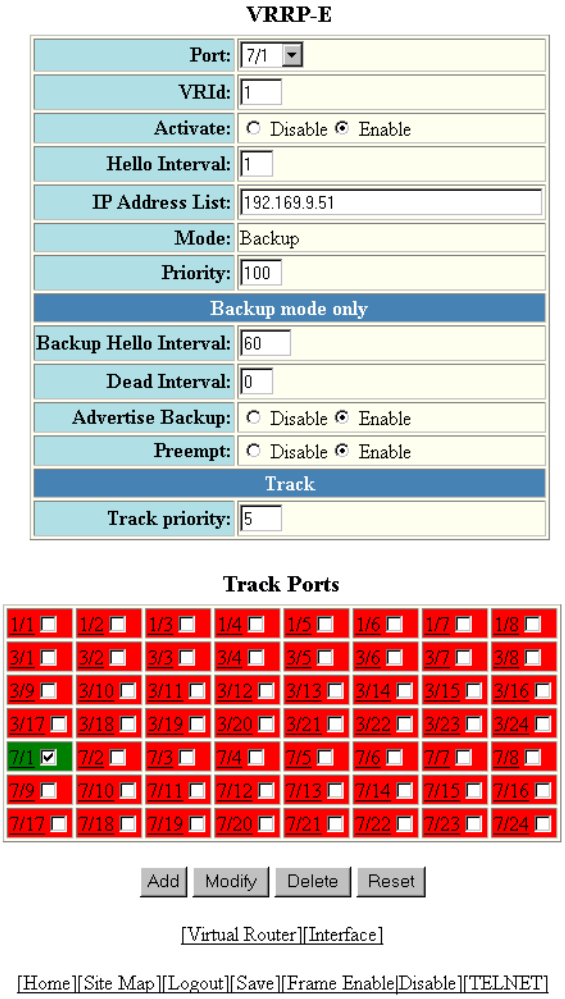
Port	VR Id	Hello Intv	Admin Status	VIP	Master Ip	Ip Address Count	Mode	State	Track Priority	Cfg Priority	Cur Priority	Backup			Track ports	
												Dead Intv	Preempt	Adv Backup		
12/1	1	1	Disable		Unknown	0	Backup	Initialize	5	100	100	0	Enable	Enable	v50,v60	Delete Modify
12/7	1	1	Disable		Unknown	0	Backup	Initialize	5	100	90	0	Enable	Enable	v50,v60	Delete Modify
Port	VR Id	Hello Intv	Admin Status	VIP	Master Ip	Ip Address Count	Mode	State	Track Priority	Cfg Priority	Cur Priority	Backup			Track ports	
												Dead Intv	Preempt	Adv Backup		

[Add virtual router]

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

- If Virtual Routers have not been configured, you see the VRRPE configuration panel (Figure 19.5):

Figure 19.5 VRRPE Configuration panel



6. Select the interface to which the configuration will be applied from the Port field.
7. Enter the VRID in the Router ID field. By default, VRID 1 is assigned to an interface.
8. By default, VRRPE is enabled for the specified interface and VRID. Click on Disable next to Activate if you want to disable VRRPE for the interface and VRID. Click on Enable to re-enable it.
9. Enter the Hello Interval or leave the field unchanged to use the default. The Hello Interval determines the number of seconds between Hello messages from the Master to the Backups for a given VRID. The interval can from 1 – 84 seconds. The default for this is 1 second.
10. Enter the IP address that this VRID is backing up.
11. Backup is always displayed for the Mode field for VRRPE.
12. Enter the backup priority value in the Priority field. Backup priority is the device's preferability for becoming the Master for the VRID. During negotiation, the router with the highest priority becomes the Master. The default Backup Priority is 100.
13. Enter the number of seconds for the Hello Interval for the backup.
14. Enter the number of seconds for the Dead Interval. This interval is the amount of time a Backup waits for a Hello message from the Master 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. The default Dead Interval is three times the Hello Interval plus one-half second

15. Select Enable for the Activate Backup field if you want to advertise routes to a backed up VRID even when the router is not the current active router for the VRID. Disabling the advertisements helps ensure that other routers do not receive invalid route paths for the VRID. The default is Disabled.
16. Select Enable for the Preempt field to prevent 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. Select Disable if you do not want to disable this feature. The default is enabled.
17. Enter the Track – Priority value or leave it blank to use the default. If a tracked port's link goes down, the VRID port's priority is reduced by the amount of the tracked port's priority. The default Track Priority value for VRRPE is 5.
18. Place a check mark in the box for a port in Track Ports section. Ports with check marks are tracked. If the link for a tracked port goes down, the VRRP priority of the VRID interface is changed, causing the devices to renegotiate for Master.

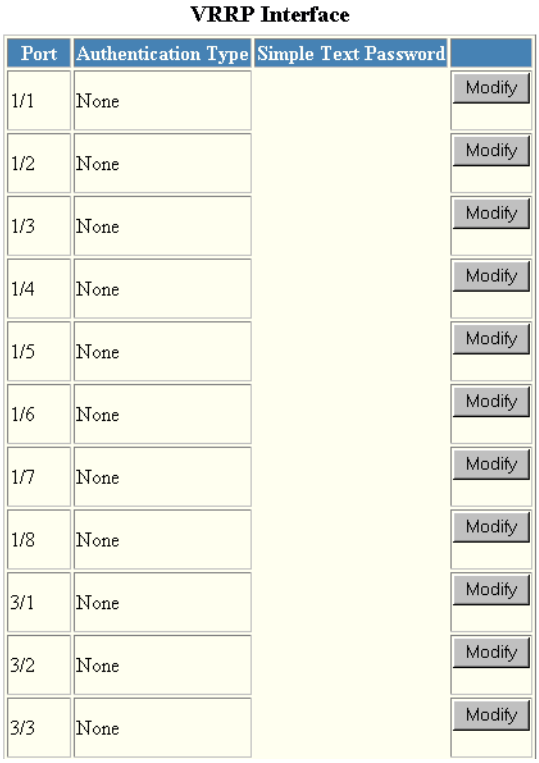
    The Track Ports area also shows which port links are up and which are down. The green color identifies ports that are up; whereas, red shows those that are down.
19. Click on the Add button to add the VRRPE port.
20. Select the Save link at the bottom of the dialog. Select Yes when prompted to save the configuration change to the startup-config file on the device's flash memory.

Modifying Interface Authentication

You can modify the password that was configured for a port on a separate panel of the Web management interface.

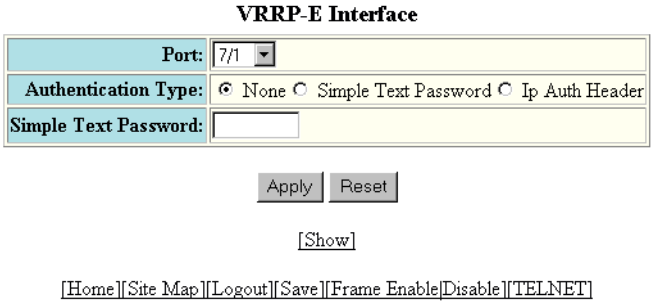
1. Log on to the device using a valid user name and password for read-write access. The System configuration dialog is displayed.
2. Make sure VRRPE is enabled.
3. Click on the plus sign next to Configure in the tree view to expand the list of configuration options.
4. Click on the plus sign next to Virtual Redundant Router to expand the list of Virtual Router options.
5. Click on the plus sign next to VRRP-Extended to display its list of options.
6. Click on the Interface link to display the VRRPE Interface table, which lists all the ports on the device.

Figure 19.6 Top of VRRPE Interface Table



7. Click on the Modify button for an interface if you want to make changes to the its authentication type and password. The VRRP-E Interface configuration panel appears.

Figure 19.7 VRRPE Interface Page



8. Select the Port number to which the configuration will apply. All the device's interfaces are listed in the pull down menu.
9. Select the Authentication Type, either None, Simple Text Password or Ip Auth header.
10. Enter a password if the authentication is Simple Text Password. Leave this field blank if other password types are used.
11. Click on the Apply button to apply your changes.
12. 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.

Displaying VRRPE Statistics

To display VRRPE statistics using the Web management interface, do the following:

1. Log on to the device using a valid user name and password for read-write access. The System configuration panel is displayed.
2. Click on the plus sign next to Monitor in the tree view to expand the list of configuration options.
3. Click on the plus sign next to Virtual Redundant Router to expand the list of Virtual Router options.
4. Click on the plus sign next to VRRP-Extended to display its list of options.
5. To display statistics for authentication used by the VRRPE, click on the Interface link.

Virtual Router Interface Statistics

Port	Header Error	Authen Type Error	Authen Password Mismatch Error	Virtual Router Id Error
12/1	0	0	0	0
12/7	0	0	0	0
Port	Header Error	Authen Type Error	Authen Password Mismatch Error	Virtual Router Id Error

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

The panel shows the following information:

Column	Description
Port	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.
Header Error	The number of VRRPE packets received by the interface that had a header error.
Authen Type Error	The number of VRRPE packets received by the interface that had an authentication error.
Authen Password Mismatch Error	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.
Virtual Router ID Error	The number of VRRP or VRRPE packets received by the interface that contained a VRID that is not configured on this interface.

1. To display statistics all other statistics for a port, click on the Virtual Router link.

VRRP-E Virtual Router

Port	VR Id	State	Receive Pkts Drop		Receive Mismatch				Rcv Priority Zero from Master	Rcv Higher Priority	Transmit Count	
			Arp	IP	Port	Num of IP	IP	Hello			Master	Backup
12/1	1	Initialize	0	0	0	0	0	0	0	0	0	0
12/7	1	Initialize	0	0	0	0	0	0	0	0	0	0
Port	VR Id	State	Receive Pkts Drop		Receive Mismatch				Rcv Priority Zero from Master	Rcv Higher Priority	Transmit Count	
			Arp	IP	Port	Num of IP	IP	Hello			Master	Backup

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

The panel shows the following information:

Column	Description
Port	Slot/port number of the interface.
VRId	The VRID for the port
State	Current state of the port. It can be: • Initialize • Master • Backup
Receive Pkts Drop	Number of packets addressed to the VRID that were dropped. Packets are divided into the following categories: • ARP packets • IP packets
Receive Mismatch	Number of packets that did not match the configured values of the following: • Port – receiving interface • IP – IP addresses • Hello – Hello interval
Receive Priority Zero from Master	Number of times the current Master has resigned
Receive Higher Priority	The number of 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.
Transition Count	The number of times this Layer 3 Switch has changed the state of its VRID: • Master – transition from Backup to Master • Backup – transition Master to Backup

Chapter 20
Route Health Injection

You can configure a Foundry Layer 3 Switch to check the health of the HTTP application and "inject" a host route into the network to force a preferred route to an actively responding web host. The web host can be directly attached to the Layer 3 Switch or can be attached through Switches. The web host can be a web server or a Foundry ServerIron or third-party Server Load Balancing (SLB) device configured with a virtual IP address (VIP) representing the HTTP application.

NOTE: This feature is supported on the Chassis Layer 3 Switches, the TurboIron/8 Layer 3 Switch, and the NetIron Stackable Layer 3 Switch.

The ***route health injection*** feature[1] enables a router to advertise a host route to a globally-distributed web site. Gateway routers that receive the host route along with other routes to the same web site in other locations can choose the best route. Web clients attached to the gateway servers thus enjoy fast response time regardless of their location, because their gateway routers use the best path to the web site. By advertising the host route instead of a network route to the web site's IP address, the Foundry Layer 3 Switch ensures that gateway routers receive a route to the IP address only if that IP address is available. The Foundry Layer 3 Switch uses a Layer-4 HTTP health check that you configure to determine whether the HTTP (web) service on the IP address is available. The health check and how to configure it are described later in this section.

NOTE: This feature supports health checks only for TCP port 80 (HTTP).

Normally, an IP address should exist on only one host on the public Internet. However, the Foundry ServerIron and some third-party SLBs allow the same IP address to exist on multiple machines using virtual IP addresses (VIPs). A VIP is an IP address that you configure on a ServerIron or third-party SLB, then associate with "real" servers attached to the ServerIron. These real servers are the web hosts that contain the web site requested by clients. In a simple SLB configuration, a single ServerIron contains a VIP that maps to multiple real servers that have identical contents. The VIP is the IP address associated with the web site on DNS servers. In a globally-distributed SLB configuration, multiple ServerIrons or other SLBs in different networks throughout the Internet are configured with same VIP and are attached to sets of real servers that contain the web site's content.

NOTE: Host-route support for globally-distributed SLB applies only to Foundry Layer 3 Switches. You can use the feature regardless of whether the Layer 3 Switch is directly attached to a web server that contains the web site or is attached to ServerIrons or third-party SLBs that load balance the IP address for the servers. Where this feature description discusses the ServerIron, the description also applies to third-party SLB products with comparable support for virtual IP addresses (VIPs).

1.This feature is also known as ***Global IP***.

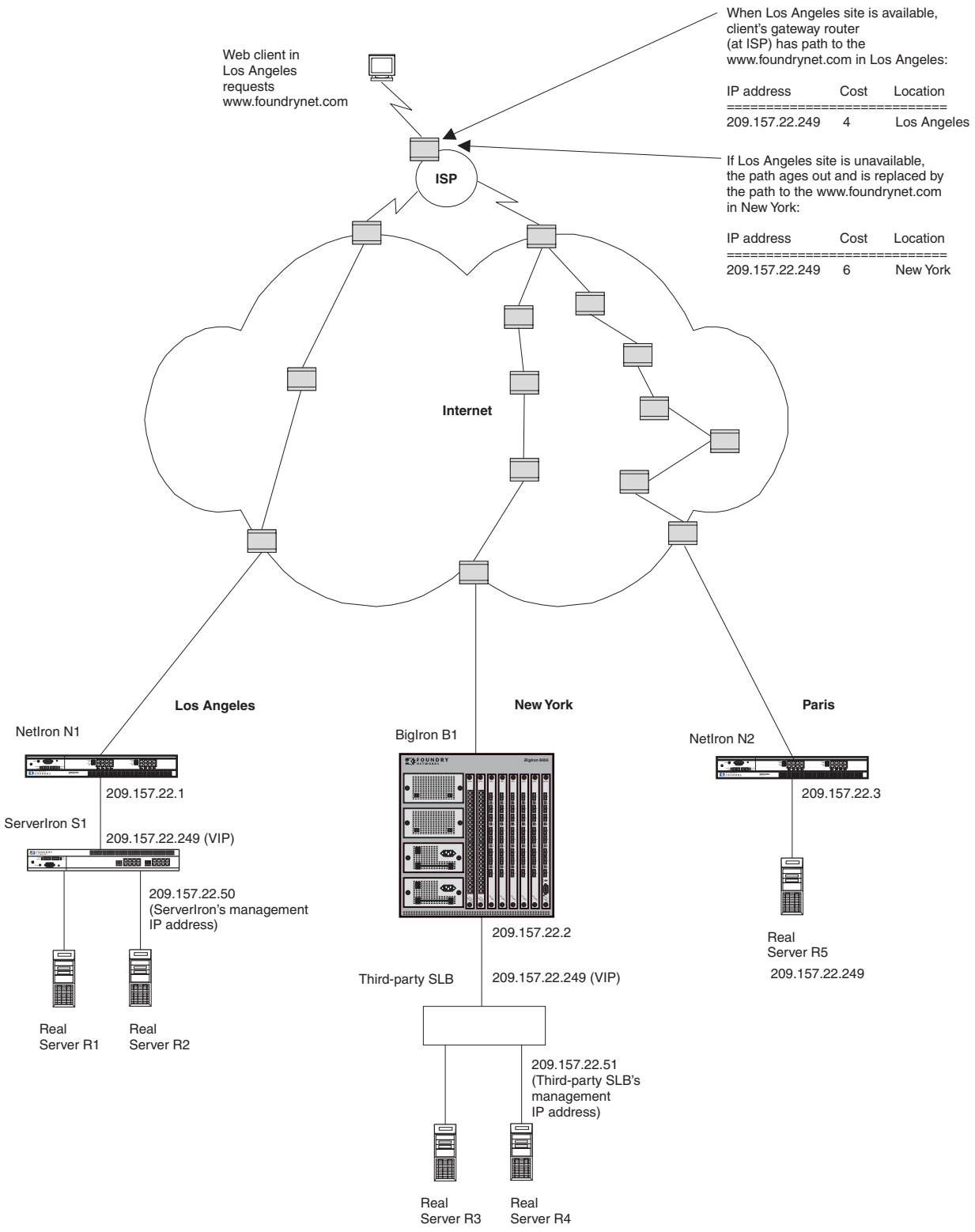
Configuration Example

Suppose you configure a ServerIron in Los Angeles and third-party SLB in New York to serve VIP 209.157.22.249. For this example, also assume that you have a real server in Paris with the same IP address and the server is directly attached to a Foundry Layer 3 Switch.

Suppose the DNS entry for this IP address maps the address to a site named www.foundrynet.com. When a web client in Los Angeles enters this domain in their web browser, the web browser goes to the client's local DNS to resolve the name into an IP address. When the DNS returns the address to the web browser, the browser then attempts to contact the HTTP port (usually TCP port 80) on the host with the IP address returned by the DNS.

Figure 20.1 shows an example of a globally-distributed SLB configuration in which the route health injection feature is used.

Figure 20.1 Route health injection configuration



When the web browser sends its TCP SYN request (to initiate the HTTP session with the web host), the gateway router used by the client's computer looks in its routing table for the route to the requested IP address. The router may receive multiple paths, in which case the router typically chooses the path with the lowest cost (usually the number of router hops to the host) to place in the routing table. The paths can all go to the same host or to different hosts. In the case of globally-distributed SLB, the paths go to different hosts. The shortest path takes the client to the gateway router attached to the ServerIron, third-party SLB, or directly-attached server that is closest to the client. Thus, when a client on the West coast requests the web site, the client's gateway sends the request to the ServerIron in Los Angeles. A client in London would instead be directed to the directly-attached server in Paris.

The router's behavior works well when all the real servers are available. However, suppose the real servers attached to the ServerIron in Los Angeles become unavailable. This results in the VIP on that ServerIron becoming unavailable.

NOTE: You can associate “remote” servers with a VIP configured on a ServerIron, in which case the ServerIron fails over to the remote servers if all local servers are unavailable. Remote servers are a less robust alternative to the ServerIron’s globally-distributed SLB feature. There is no restriction against using remote servers in a globally-distributed SLB configuration, but globally-distributed SLB makes the use of remote servers unnecessary.

In a globally-distributed SLB configuration, a client can still reach the desired VIP (web site) if the client's gateway router receives a path to another site that contains the VIP the client is trying to reach. However, gateway routers typically advertise network routes rather than host routes. As a result, even if the VIP (web site) is unavailable, the gateway router still advertises the network to which the VIP belongs. Consequently, a client's gateway router can still have a path to the unavailable server, in which case the client does not receive the requested web page.

By configuring the Foundry Layer 3 Switches attached to the ServerIrons, third-party SLBs, or real servers that contain the web site to check the health of the web site (HTTP application), you can ensure that the Foundry Layer 3 Switches advertise paths only to for web site locations that are available:

- If the web site passes the health check, the Foundry Layer 3 Switch advertises a host route to the web site’s IP address.
- If the web site fails the health check, the Foundry Layer 3 Switch removes the host route. The route is no longer advertised and ages out of the routing tables in clients’ gateway routers.

As a result, those paths to the web site’s IP address that are no longer available age out of the routing tables on gateway routers while the paths that are still available remain in the routing tables. When a client uses its gateway router to reach the web site, the gateway's path to the site’s IP address is usually the one with the lowest cost. In Figure 20.1, when the site at Los Angeles is available, the client's gateway uses the path to Los Angeles as the route to IP address 209.157.22.249. However, if the IP address at the Los Angeles site becomes unavailable and thus fails its health check, the NetIron at the Los Angeles site removes the static host route for 209.157.22.249 from its route table. The path on the client's gateway ages out and is replaced by the next valid path with the lowest cost, in this case the path to 209.157.22.249 at the New York site.

HTTP Health Check Algorithm

When you configure a Layer 3 Switch to periodically check the health of the HTTP port on a web server, the router does one of the following based on the result of the health check. The health check algorithm applies regardless of whether the web server is directly attached to the Layer 3 Switch (or attached through Switches) or is attached to a ServerIron or third-party SLB that is load balancing the IP address among multiple servers.

- If the health check is successful, the router places a static host route in its route table for the web site’s IP address. When the router sends a routing advertisement, the host route is included. The client's gateway router will receive this host route as one of the paths to the IP address.
- If the health check is not successful, the router removes the static host route (if present) for the IP address. As a result, the route ages out of the routing tables on other routers. After the removed route ages out of the routing table on the client's gateway router, the router accepts another path to the IP address.

You can configure a separate HTTP health check for each web site IP address. The health check consists of a standard TCP connection followed by a standard request for an HTTP page on the IP address. If the HTTP page responds with an acceptable HTTP status code, the IP address passes the health check, at which point the

Foundry Layer 3 Switch leaves the static host route to the IP address in the Layer 3 Switch's route table or adds the route if it is not present.

By default, the HTTP health check is disabled. Once you enable the health check, the Layer 3 Switch sends the health check every five seconds by default. The default health check consists of a HEAD request for the default home page "1.0". If the web site does not respond to a health check, the Layer 3 Switch resends the health check up to two more times by default before determining that the web site is no longer available and removing the static host route for the web site.

All the health check parameters are configurable. See "CLI Syntax" on page 20-5.

Configuration Considerations

- The Foundry Layer 3 Switch and the ServerIron, third-party SLB, or real server must be in the same IP subnet.
- Place the management station for the NetIron on a different subnet than the one that contains the web site (HTTP application) whose health you are checking. If the web site and the management station are on the same subnet, the **ip dont-advertise** command (see "CLI Syntax" on page 20-5) will prevent you from reaching the NetIron through the management station.
- You cannot use the same Layer 3 Switch port for OSPF and for the globally-distributed SLB. If the port already contains configuration information for one of these features, you cannot configure the other feature unless you first remove the configuration information for the first feature.

CLI Syntax

Use the following commands to configure the health check parameters on the Layer 3 Switch.

Global CONFIG Level

Use the following command at the global CONFIG level to identify the VIP that has the HTTP port the Layer 3 Switch is checking.

Syntax: server real <name> <vip>

The <name> parameter identifies the ServerIron, third-party SLB, or real server. This value does not need to match a value on the ServerIron, third-party SLB, or real server. The value simply identifies the ServerIron, third-party SLB, or real server uniquely on the Layer 3 Switch.

The <vip> parameter is the IP address of the web site. If the web server is directly attached to the Layer 3 Switch, this is the IP address of the web server. If the web server is attached to a ServerIron or third-party SLB, the VIP is the virtual IP address configured on the ServerIron or third-party or SLB for the web site.

Use the following commands to change the interval and retry values for the HTTP health check. When you press Enter after the first command, the CLI changes to the TPC/UDP port configuration level for port 80.

Syntax: server port 80

Syntax: tcp keepalive <interval> <retries>

The <interval> parameter specifies the number of seconds between health checks sent by the Layer 3 Switch. You can specify a number from 2 – 60 seconds. The default is 5 seconds.

The <retries> parameter specifies how many times the Layer 3 Switch will resend a health check if the web site does not respond. You can specify from 1 – 5 retries. The default is 2.

Real Server Level

After you enter the **server real…** command shown above, the CLI changes to the Real Server level.

The following command enables the HTTP health check for the web site. The health check is disabled by default.

Syntax: port http keepalive

The following command is optional and changes the default method and URL for the health check. By default, the Layer 3 Switch sends a HEAD request for the default homepage, "1.0". The slash in the URL is optional; the Layer 3 Switch inserts the slash for you if you leave it out.

Syntax: port http url "[GET | HEAD] [/]<URL-page-name>"

The following command changes the HTTP status codes that the Layer 3 Switch accepts as valid responses to a health check. The default status code range for HTTP health checks in SLB configurations is 200 – 299. You can specify up to four discrete ranges. To specify a single message code for a range, enter the code twice. For example to specify 200 only, enter the following command: **port http status_code 200 200**.

Syntax: port http status_code <range> [<range>[<range>[<range>]]]

Interface Level

The following commands configure an IP subnet address that is in the same subnet as the web site's IP address. Enter these commands on the interface that connects the Layer 3 Switch to the real server or to the ServerIron or third-party SLB that is load balancing for the IP address.

The **ip dont-advertise** command configures the Layer 3 Switch to block advertisement of the network on the interface. If you do not block advertisement of the network, the Layer 3 Switch will advertise a route to the network containing the web site even if the web site itself is unavailable. After you enter the **ip dont-advertise** command, the Layer 3 Switch advertises only a host route to the IP address. Thus, if the web site fails the HTTP health check, the Layer 3 Switch removes the static host route for the web site's IP address and also does not advertise a network route for the network containing the IP address.

Syntax: ip address <ip-addr> <ip-mask> [secondary]

Or

Syntax: ip address <ip-addr>/<mask-bits>

Syntax: ip dont-advertise <ip-addr> <mask>

Or

Syntax: ip dont-advertise <ip-addr>/<mask-bits>

NOTE: An IP address in the subnet you want to block must already be configured on the interface.

Configuring the HTTP Health Check on the Layer 3 Switch

To configure Foundry Layer 3 Switches to perform the HTTP health check for a web site and to manage a static host route for the IP address, do the following:

- Identify the web site's IP address on the Layer 3 Switch.
- Enable the HTTP keepalive (health check).
- Optionally modify the health-check keepalive interval and retries.
- Optionally modify site-specific health check parameters (the URL requested by the health check and the HTTP status codes that the Layer 3 Switch will accept as a normal response).
- Configure the port that connects the Layer 3 Switch to the HTTP application (ServerIron, third-party SLB, or real server) to not advertise the network route for the IP subnet the ServerIron, third-party SLB, or real server and the port are on.

For example, to configure Foundry Layer 3 Switches for the configuration shown in Figure 20.1, enter the following CLI commands.

CLI Commands for NetIron N1

To configure the health check on NetIron N1, enter the following commands:

```
NetIron(config) server real S1 209.157.22.249
NetIron(config-rs-S1) port http keepalive
NetIron(config-rs-S1) interface ethernet 6
NetIron(config-if-6) ip address 209.157.22.1/24
NetIron(config-if-6) ip dont-advertise 209.157.22.1/24
NetIron(config-if-6) write memory
```

Syntax: server real <name> <ip-addr>

Syntax: port http keepalive

Syntax: ip dont-advertise <ip-addr> <mask>

Or

Syntax: ip dont-advertise <ip-addr>/<mask-bits>

The **server real** command in this example configures the NetIron to send an HTTP health check to the HTTP port on IP address 209.157.22.249. When you press Enter after this command, the CLI changes to the Real Server level of the CLI. This level allows you to configure health check parameters for the HTTP port on the IP address.

The **port http keepalive** command in this example is entered at the Real Server level and enables the HTTP health check. The health check is disabled by default, so you must enter this command. You can enter additional commands at this level to modify the health check parameters. These commands are shown in the examples for BigIron B1 and NetIron N2.

The **ip address** command adds an IP interface for the connection to the IP address. This interface must be in the same subnet as the IP address.

NOTE: To configure the ServerIron itself for multiple IP subnets, add IP addresses using the **source-ip** command. See the *Foundry ServerIron Installation and Configuration Guide* for information.

The **ip dont-advertise** command configures the Foundry Layer 3 Switch to block advertisement of the network route for this IP subnet address. This command ensures that the Layer 3 Switch advertises only the host route to the IP address. If the Layer 3 Switch advertises the network route to the subnet containing the IP address, then even if the Layer 3 Switch removes the host route from its routing table, it will still advertise the network route to the IP address (and thus to the web server), defeating the failover capability of globally-distributed SLB.

CLI Commands for BigIron B1

The following commands configure BigIron B1 for the configuration shown in Figure 20.1 on page 20-3.

```
BigIron(config) server real S2 209.157.22.249
BigIron(config-rs-S2) port http keepalive
BigIron(config-rs-S2) port http url "/sales.html"
BigIron(config-rs-S2) port http status_code 200 205
BigIron(config-rs-S2) interface ethernet 1/3
BigIron(config-if-1/3) ip address 209.157.22.2/24
BigIron(config-if-1/3) ip dont-advertise 209.157.22.2/24
BigIron(config-if-1/3) write memory
```

Syntax: port http url “[GET | HEAD] [/]<URL-page-name>”

Syntax: port http status_code <range> [<range>[<range>[<range>]]]

The **port http url** command changes the URL that the Layer 3 Switch sends as part of the health check. By default, the Layer 3 Switch sends an HTTP HEAD request for the default page (“1.0”). If you enter a URL, the health check instead requests that URL. The slash (/) is an optional parameter. If you do not set the GET or HEAD parameter, and the slash is not in the configured URL page, then the Layer 3 Switch automatically inserts a slash before retrieving the URL page.

In addition to specifying another URL, you can change the method to GET. Changing the method does not affect the health check from the Layer 3 Switch's standpoint. You can use either method.

The **port http status_code** command in this example changes the range of HTTP status codes the Layer 3 Switch accepts as normal (healthy) replies to a health check.

CLI Commands for NetIron N2

The following commands configure NetIron N2 for the configuration shown in Figure 20.1. This example includes the commands for modifying the HTTP health check interval and retry values.

```
NetIron(config) server port 80
NetIron(config-port-80) tcp keepalive 10 3
NetIron(config-port-80) server real S3 209.157.22.249
NetIron(config-rs-S2) port http keepalive
NetIron(config-rs-S2) port http url "/marketing.html"
NetIron(config-rs-S2) interface ethernet 9
NetIron(config-if-9) ip address 209.157.22.3/24
NetIron(config-if-9) ip dont-advertise 209.157.22.3/24
NetIron(config-if-9) write memory
```

Syntax: server port 80

Syntax: tcp keepalive <interval> <retries>

The <interval> parameter specifies the number of seconds between health checks sent by the Layer 3 Switch. You can specify a number from 2 – 60 seconds. The default is 5 seconds.

The <retries> parameter specifies how many times the Layer 3 Switch will resend a health check if the web site does not respond. You can specify from 1 – 5 retries. The default is 2.

Displaying Server and Application Port Information

You can use the CLI to display the following types of information:

- Server (virtual IP address) information
- Application port information

Displaying Server Information

To display information about the server virtual IP addresses (VIPs) you have configured, enter a command such as the following at any level of the CLI:

```
BigIron# show server real RS2

Real Servers Info

Server State - 1:enabled, 2:failed, 3:test, 4:suspect, 5:grace_dn, 6:active
Name:RS2             IP:  209.157.23.60:4    State:1
```

Syntax: show server real <name>

This display shows the following information.

Table 20.1: Real Server Information

This Field...	Displays...
Server State	The possible values for the server state. The state of each real server is shown by the State field. See below.
Name	The name of the real server. This is the name you assigned to the server when you configured it on the ServerIron.
IP	The IP address of the real server. If you configured a host range of VIPs on the server, the number following the IP address (after the colon) is the number of hosts on the server.
State	The state of the real server. The state can be one of the states listed by “Server State” at the top of the display.

Displaying Keepalive Information

To display the keepalive parameters in effect for the application ports on the servers, enter the following command at any level of the CLI:

Syntax: show server keepalive-port

Chapter 21
Configuring FSRP

This chapter describes how to configure the Foundry Standby Router Protocol (FSRP), a Foundry Networks proprietary protocol that provides redundant paths between two routers.

Details for configuring FSRP with the CLI and the Web management interface are shown. For detailed summaries of all CLI commands, including the syntax and ranges of parameter values, see the *Foundry Switch and Router Command Line Interface Reference*.

For information about the differences between VRRP and FSRP, see "Differences Between FSRP and VRRP" on page 21-5.

NOTE: Beginning with release 07.6.01, this feature is not supported in Layer 3 Switch images.

Overview of Foundry Standby Router Protocol (FSRP)

FSRP allows alternate paths to be provided to a host. To provide path redundancy between given hosts, a ***virtual router*** with its own unique IP addresses is created. The virtual router is created by assigning these unique IP addresses to ports on existing routers in the network—routers that could provide a path between the given hosts.

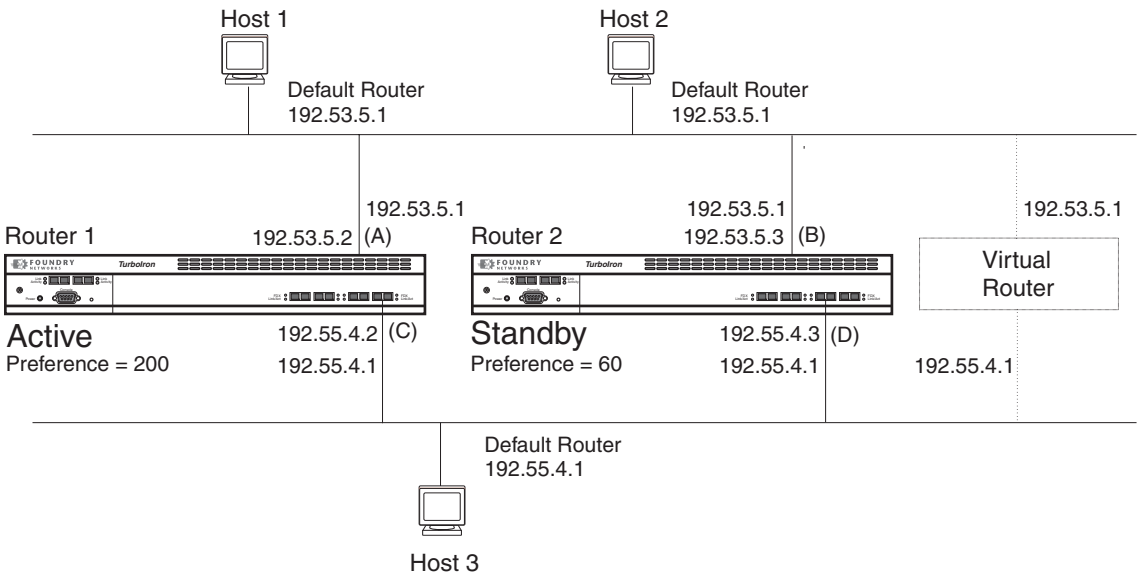
NOTE: Virtual IP router addresses are in addition to the IP address assigned to each IP interface.

For example, in Figure 21.1, suppose you want to provide continual connectivity between Host 1 and Host 3 with the use of redundant paths. A virtual router is created by assigning the same virtual router IP address to all physical interfaces that will provide redundant paths for that portion of the network. Virtual router IP address 192.53.5.1 is assigned to interfaces A and B, and the virtual router IP address 192.55.4.1 is assigned to interfaces C and D. Notice that in both cases, these virtual addresses are in addition to their physical IP addresses.

The virtual IP address also serves as the default router for the hosts. Hosts 1 and 2 reference the virtual IP router address 192.53.5.1 as their default router and Host 3 references the virtual router IP address, 192.55.4.1.

If Router 1 goes down, then Router 2 provides connectivity between Host 1 and Host 3.

Figure 21.1 FSRP operating in a NetIron network



Router 1:
IP Address for Interface A: 192.53.5.2
Virtual Router IP address for Interface A: 192.53.5.1
IP Address for Interface C: 192.55.4.2
Virtual Router IP address for Interface C: 192.55.4.1

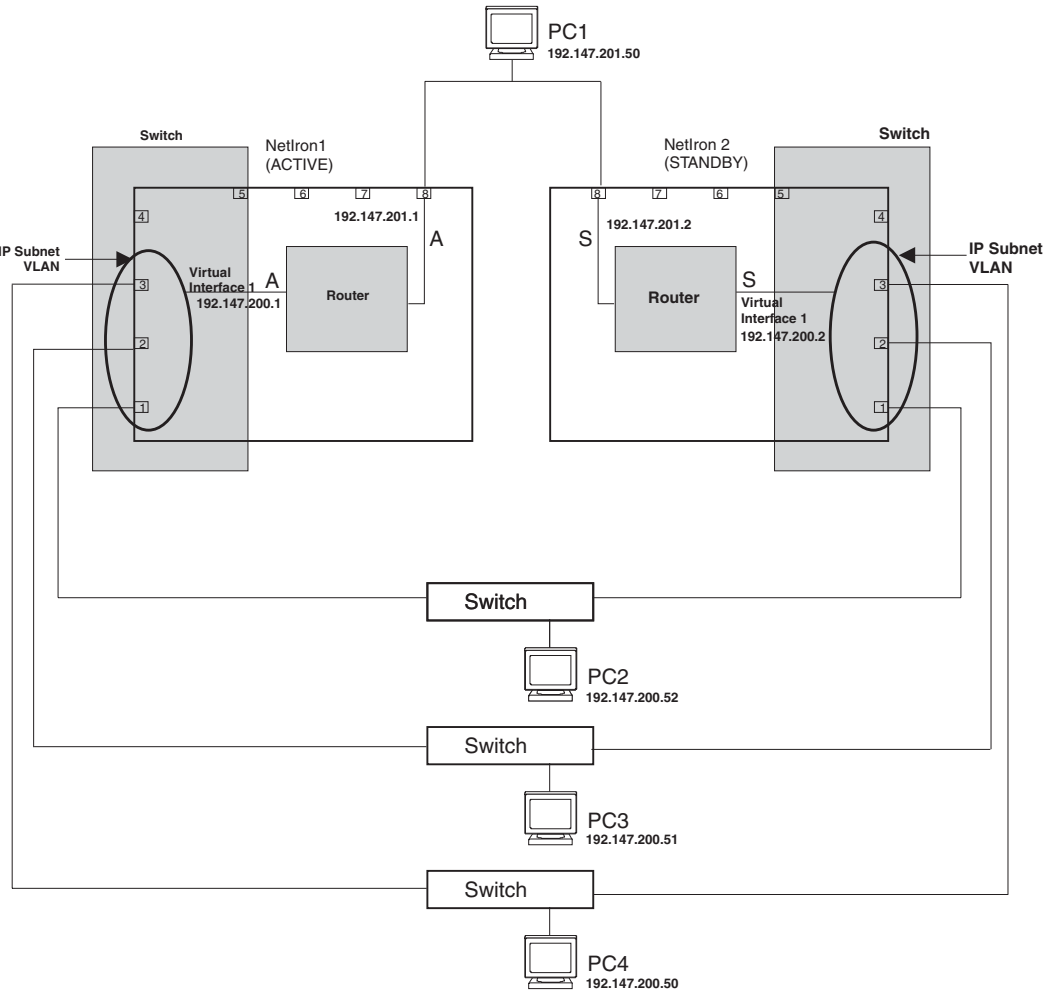
Router 2:
IP Address for Interface B: 192.53.5.3
Virtual Router IP address for Interface B: 192.53.5.1
IP Address for Interface D: 192.55.4.3
Virtual Router IP address for Interface D: 192.55.4.1

FSRP Support on Virtual Interfaces

FSRP is supported on both physical and virtual interfaces. Support on a virtual interface allows you to assign a single virtual interface to serve as a redundant link for multiple ports within a VLAN. For example, in Figure 21.2, virtual interface 1 represents ports 1, 2, and 3 for NetIron1.

A virtual interface will by default remain active until all underlying links go down. If you want the virtual link to go to FSRP standby state when a subset of the ports goes down, you must configure track ports as well.

Figure 21.2 Virtual interface providing a redundant link



Active and Standby Routers

To establish one router as active, you assign a higher preference to the router. If the preference for two routers is equal, the interface with the higher IP address takes precedence as the active router. Link status is monitored using a track port.

Track Ports

A ***track port*** tracks the status of the ports that are providing redundant paths. You can assign any port to be a track port; however, a port that is providing a redundant path cannot serve as its own track port. A track port should be assigned to track each port that is part of a virtual link. For example, in Figure 21.1, interfaces A, B, C, and D should all be assigned track ports.

If a change in state (up or down) is detected by the track port, the priority of the FSRP Group Interface will automatically be increased or decreased.

NOTE: Virtual router interfaces cannot be assigned as track ports.

Multiple Track Port Support

You can assign multiple ports to serve as track ports for FSRP redundant links. If an active link fails, all FSRP interfaces that serve as track ports for the failed link are placed in standby mode.

This feature allows you to configure a system so that a given router and its defined redundant links will be in either active or standby mode. Multiple track port assignment prevents a mix of active and standby links to exist on a router.

For example, in Figure 21.3, links on NetIron1 designated as e1 and e3 have failed and have transferred control to their standby links on NetIron2; e4 and e2 remain as active links. This results in NetIron1, the router that was originally assigned to serve as the active router, having a mix of active and standby links.

To bias all traffic and link traffic to the standby router, assign all other redundant links as track ports for all other interfaces on the router. For example, on NetIron1, you would assign interfaces e1, e2, and e3 as track ports for e4. Interfaces e1, e2, and e4 would thus track port e3. Interfaces e2, e3, and, e4 would track port e1. Interfaces e1, e3, and e4 would track port e2. Configured in this manner, a failure on NetIron1 links e1 and e3 would make NetIron2 the active router for all the links seen in Figure 21.4.

Because one router and all its links are active and the other router and its links are all in standby mode, all traffic will be directed to the active router.

Figure 21.3 Failure of e1 and e3 links results in mixed active and standby links on NetIron1 without the use of multiple track ports

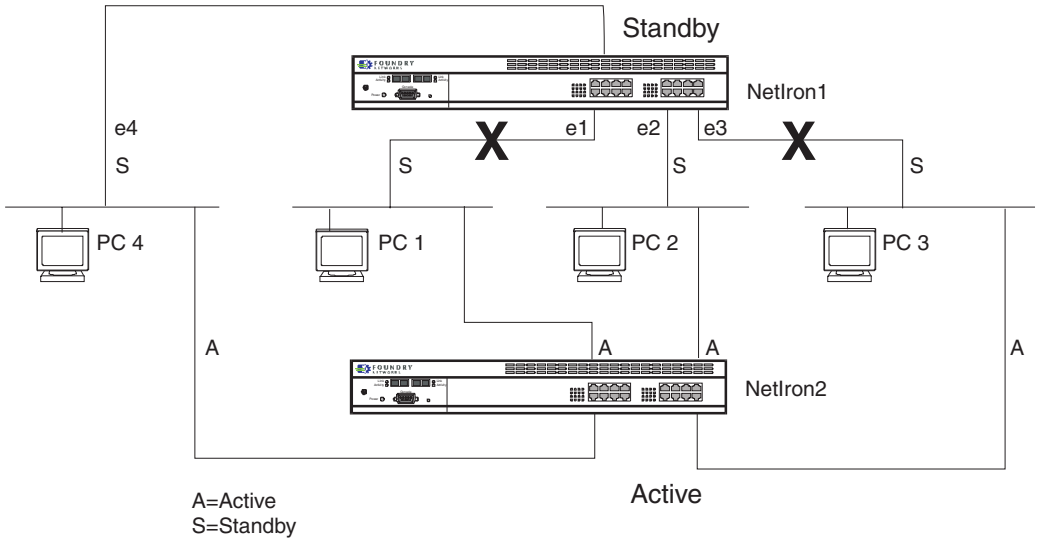
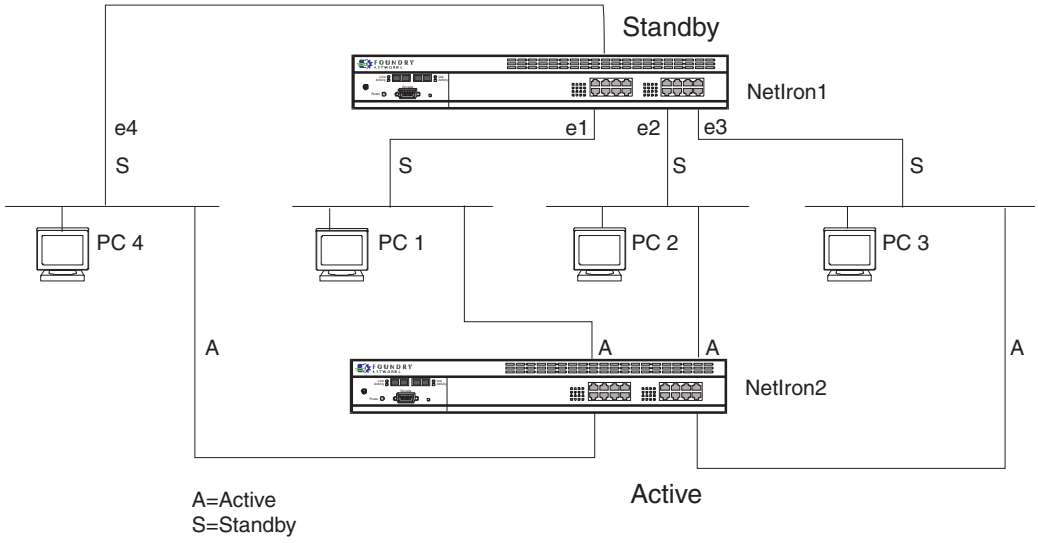


Figure 21.4 NetIron2 becomes active router after links e1 and e3 fail with multiple track ports defined



Independent Operation of RIP and OSPF

FSRP operation is independent of the RIP and OSPF protocols. RIP and OSPF operation will be unaffected when FSRP is enabled on its interfaces.

Dynamic FSRP Configuration

All FSRP global and interface parameters are dynamically activated. You do not need to reset the system to place FSRP configuration parameters into effect.

Differences Between FSRP and VRRP

The Virtual Router Redundancy Protocol (VRRP) is a standards-based protocol that provides redundancy to routers within a LAN. VRRP is described in RFC 2338. Foundry's implementation of VRRP provides many of the same features as FSRP. In addition, VRRP enables you to configure third-party devices that adhere to RFC 2338 along with Foundry devices as virtual routers. FSRP requires that the other devices support FSRP.

If you are configuring Foundry routers for redundancy, you can use either protocol. The features provided by the two protocols are similar, yet the protocols do differ in the following ways:

- VRRP uses an IP multicast address for VRRP management traffic, while FSRP uses pre-defined unicast addresses.
- VRRP uses real IP addresses assigned to an interface and does not use virtual IP addresses, whereas FSRP must use one pre-defined virtual IP address for each virtual router. You can associate a VRRP virtual router with an IP address or with a virtual interface (a named set of physical interfaces).
- Each VRRP virtual router (denoted by a unique Virtual Router ID [VRID]) can have one Master router and one or more Backup routers. In contrast, each FSRP router can have one Primary Router and only one Standby Router. Most VRRP and FSRP configurations consist of two routers—one active router (Master or Primary) and one standby router (Backup or Standby).
- Foundry's implementation of VRRP supports authentication using simple clear text passwords. FSRP does not support authentication.

NOTE: If your Foundry routers already are using FSRP and you do not need redundancy with devices that cannot use FSRP, you do not need to reconfigure your routers to use VRRP.

Foundry Networks recommends that you do not use VRRP and FSRP on the same device.

Configuring FSRP

To begin using FSRP on the router:

1. Enable operation of FSRP on the router.
2. Configure FSRP parameters on physical or virtual interfaces for those IP subnets for which a redundant path is desired. Configure the virtual router IP address and the other router's IP address.
3. Assign track ports, if appropriate.
4. Assign one of the routers to serve as the active router using the preference parameter, as appropriate.
5. Modify interface parameters, keep-alive-time, and router-dead-interval on both routers as required.

NOTE: You initially enable FSRP at the global CONFIG level of the CLI using the **router fsrp** command. All other parameters are assigned or modified at the interface level of the CLI using **ip fsrp address** <ip-addr> [<parameter>] commands.

NOTE: If you are using the Web Management interface, you enable FSRP on the System configuration panel. All other parameters (interface) are configured on the FSRP configuration panel.

Configuration Rules for FSRP

- Virtual interfaces cannot be assigned as track ports.
- The keep-alive-time value must be set to the same value on both the active and standby router when both routers are connected to the same subnet.
- The router-dead-time parameter must be set to the same value on both the active and standby routers when both routers are connected to the same subnet.

Enable FSRP on the Router

Before configuring FSRP to provide redundancy for a router, you must enable the feature on the router.

USING THE CLI

To enable FSRP on a router, enter the following command:

```
NetIron(config)# router fsrp
```

USING THE WEB MANAGEMENT INTERFACE

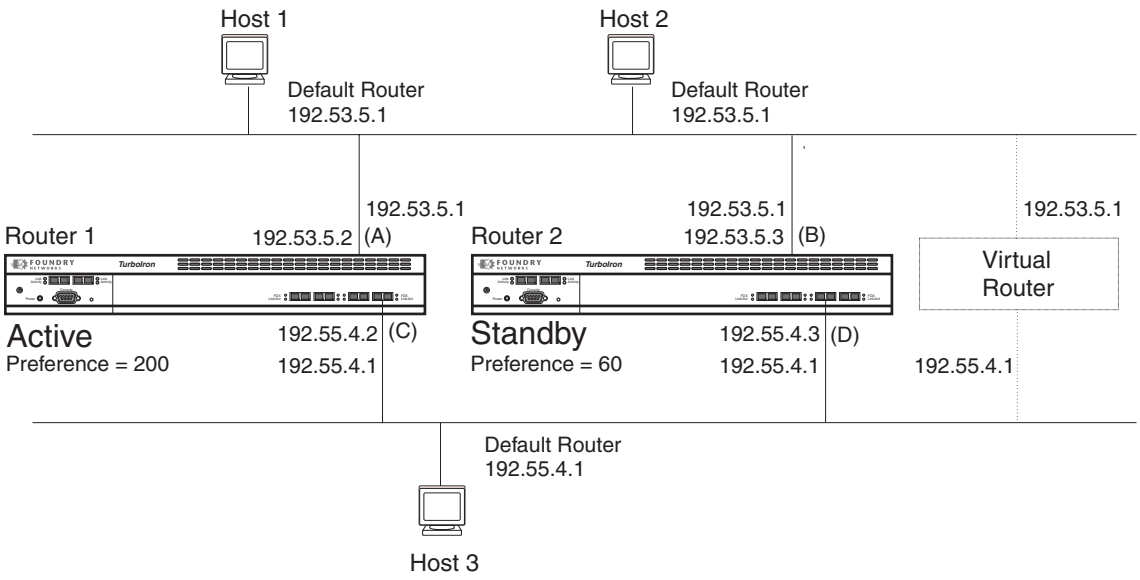
1. Log on to the device using a valid user name and password for read-write access. The System configuration dialog is displayed.
2. Select the Enable radio button next to FSRP.
3. Click the Apply button to apply the change to the device's running-config file.
4. Select the Save link at the bottom of the dialog. Select Yes when prompted to save the configuration change to the startup-config file on the device's flash memory.

NOTE: All FSRP configurations are implemented using a single configuration panel of the Web management interface. Given this, all other configuration steps, other than enabling the feature, are shown in a separate section at the end of this chapter rather than interspersed with CLI examples.

Assign Virtual Router IP Addresses

In the examples in this section, FSRP is used to provide a redundant path between Host 1 and Host 3 to ensure against failure of the primary path. See Figure 21.5.

Figure 21.5 FSRP operating in a NetIron network



Router 1:
IP Address for Interface A: 192.53.5.2
Virtual Router IP address for Interface A: 192.53.5.1
IP Address for Interface C: 192.55.4.2
Virtual Router IP address for Interface C: 192.55.4.1

Router 2:
IP Address for Interface B: 192.53.5.3
Virtual Router IP address for Interface B: 192.53.5.1
IP Address for Interface D: 192.55.4.3
Virtual Router IP address for Interface D: 192.55.4.1

USING THE CLI

EXAMPLE:

To define and assign the virtual router IP addresses for NetIron1, shown in Figure 21.5, you would need to define two separate virtual IP addresses for interfaces A and C and link those addresses to the IP addresses of the physical interfaces for A and C.

This example assumes that interface A corresponds to physical interface 17, and interface C corresponds to physical interface 20.

Configuring Router 1

To establish the virtual IP address 192.53.5.1 for interface A defined by IP address 192.53.5.2 and Ethernet port 17, enter the following commands:

```
Router1(config)# inter e 17
```

```
Router1(config-if-17)# ip fsrp address 192.53.5.2 vir-rtr-ip 192.53.5.1 other-rtr-ip
192.53.5.3
```

Notice that the latter command also defines the other router used in this configuration by entering the IP address for Interface B on Router 2 (**other-rtr-ip 192.53.5.3**).

To establish the virtual IP address 192.55.4.1 for interface C defined by IP address 192.55.4.2 and Ethernet port 20, enter the following commands:

```
Router1(config)# inter e 20
```

```
Router1(config-if-20)# ip fsrp address 192.55.4.2 vir-rtr-ip 192.55.4.1 other-rtr-ip
192.55.4.3
```

Notice that the latter command also defines the other router used in this configuration by entering the IP address for Interface D on Router 2 (**other-rtr-ip 192.55.4.3**).

Configuring Router 2

To define and assign the virtual router IP address for Router 2, you would need to define two separate virtual IP addresses for interfaces B and D as well as linking those address to the IP addresses of the physical interfaces for A and C.

This example assumes that interface B corresponds to physical interface 17, and interface D corresponds to physical interface 22.

To establish the virtual IP address 192.53.5.1 for interface B defined by IP address 192.53.5.3 and Ethernet port 17, you would enter the following commands. Note that you also are defining the other router used in this configuration by entering the IP address for interface A on Router 1 (**other-rtr-ip 192.53.5.2**).

```
Router2(config)# inter e 17
Router2(config-if-17)# ip fsrp address 192.53.5.3 vir-rtr-ip 192.53.5.1 other-rtr-ip
192.53.5.2
```

NOTE: The steps outlined in examples 1 and 2 also should be followed when creating and assigning the virtual router IP address 192.55.4.1 for interfaces C (192.55.4.2) and D (192.55.4.3).

Assign the Track Port(s)

Track ports monitor the relationship between the active and standby routers.

EXAMPLE:

To assign interface 1 to act as the track port for interface A (e17) on Router 1, enter the following commands:

```
Router1(config)# inter e 17
Router1(config-if-17)# ip fsrp address 192.53.5.2 track 1
```

NOTE: The IP address referenced in the track port assignment command is the IP address of the physical interface.

NOTE: The track port can also be assigned when assigning the virtual router IP address, as an extension to that command.

Assigning the Active Router

To establish one router as active, assign it a higher ***preference level***. If the preference level for the two routers is equal, the interface with the higher IP address takes precedence as the active router.

EXAMPLE:

To make Router 1 the active router, assign a preference value to interfaces A and C that is higher than the preference value of interfaces B and D on Router 2.

To assign a preference value of 200 to interfaces A and C, you would enter the following commands:

```
Router1(config)# int e 17
Router1(config-if-17)# ip fsrp address 192.53.5.2 preference 200
Router1(config-if-17)# int e 20
Router1(config-if-20)# ip fsrp address 192.55.4.2 preference 200
```

Modify Port Parameters (optional)

The user can also modify two port parameters for FSRP: the keep-alive-time and the router-dead-interval.

Keep Alive Time

The ***keep-alive-time*** parameter allows you to modify how often the FSRP hello message is sent on the interface on which the keep-alive-time is configured.

EXAMPLE:

To modify the keep-alive-time parameter for interfaces A and C on Router 1 to 15 seconds from the default of 3 seconds, enter the following:

```
Router1(config)# int e 17
Router1(config-if-17)# ip fsrp 192.53.5.2 keep-alive-time 15
Router1(config-if-17)# int e 20
Router1(config-if-20)# ip fsrp 192.55.4.2 keep-alive-time 15
```

NOTE: The keep-alive-time value must be set to the same value on both the active and standby routers when both routers are connected to the same subnet.

Router Dead Time

The ***router-dead-time*** parameter allows you to define the period of time (hold time) that the standby router waits before determining that the active router is unavailable (dead). If the configured period of time expires, the standby router becomes active.

NOTE: The router-dead-time parameter must be set to the same value on both the active and standby router when both routers are connected to the same subnet.

EXAMPLE:

To modify the router-dead-time parameter for interfaces A and C on Router 1 to 30 seconds from the default of 9 seconds, you would enter the following:

```
Router1(config)# int e 17
Router1(config-if-17)# ip fsrp 192.53.5.2 router-dead-interval 30
Router1(config-if-17)# int e 20
Router1(config-if-20)# ip fsrp 192.55.4.2 router-dead-interval 30
```

USING THE WEB MANAGEMENT INTERFACE

EXAMPLE:

To define and assign the virtual router IP addresses for Router 1, shown in Figure 21.5, you would need to define two separate virtual IP addresses for interfaces A and C as well as linking those address to the IP addresses of the physical interfaces for A and C.

For purposes of this example we are assuming that interface A corresponds to physical interface 17 and interface C corresponds to physical interface 20.

To enable FSRP on an interface:

1. Log on to the device using a valid user name and password for read-write access. The System configuration dialog is displayed.
2. Click on the plus sign next to Configure in the tree view to expand the list of configuration options.
3. Click on the FSRP link.
   - If the device does not have an FSRP configuration, the FSRP configuration panel is displayed.
   - If FSRP is already configured but you are adding a new FSRP configuration, click on the Add Interface link to display the FSRP configuration panel, as shown in the following example.
   - If you are modifying an existing FSRP configuration, click on the Modify button to the right of the row describing the configuration to display the FSRP configuration panel.

4. Select the IP address to be configured from the IP Address field's pull down menu. For example, if you are initially assigning FSRP to interface A (Router 1) as shown in Figure 21.5, you would select IP address 192.53.5.2.
5. Assign a virtual IP address for the virtual router. A virtual router IP address needs to be configured on at least one router in the FSRP group. For interface A, you would assign 192.53.5.1, as shown in the network configuration of Figure 21.5.

NOTE: The default IP address for a virtual router is 0.0.0.0.

6. Enter the other router IP address. This is the physical IP address of the partner router's interface in the active-standby router relationship. Notice that in the case of the example (Figure 21.5), interface B on router 2 is designated as the standby router interface so IP address 192.53.5.3 is entered.
7. To establish a router as the active router in the redundancy configuration, a higher value should be entered for its preference level. In this case, because router 1 is the desired active router and the router currently being configured, a value of 200 is entered.
8. Modify the keep alive time parameter if a value other than the default value of 3 seconds is desired. For this configuration, modify the value to 15.

NOTE: The keep alive time parameter allows the user to modify how often the FSRP hello message is sent on an interface. Possible values are 1 – 120 seconds. The default is 3 seconds.

NOTE: The keep alive time parameter must be set to the same value on both the active and standby routers when both routers are connected to the same subnet.

9. Modify the dead time parameter if a value other than the default value of 9 seconds is desired. For this configuration you would modify the value to 30.

NOTE: The dead time parameter allows you to define the period of time (hold time) that the standby router will wait before determining that the active router is unavailable (dead). When the configured period of time expires, the standby router will become active. Possible values are 3 – 255. The default value is 9 seconds.

NOTE: The dead time parameter must be set to the same value on both the active and standby routers when both routers are connected to the same subnet.

10. Select the track port by selecting a box next to the desired interface. For purposes of this example, you would select interface 1 as the track port for interface A on router 1.

NOTE: The track port is a physical port that is used to track the status of ports that provide redundant paths. If the software detects a change in state (up or down), the software increases or decreases the priority of the FSRP Group Interface accordingly.

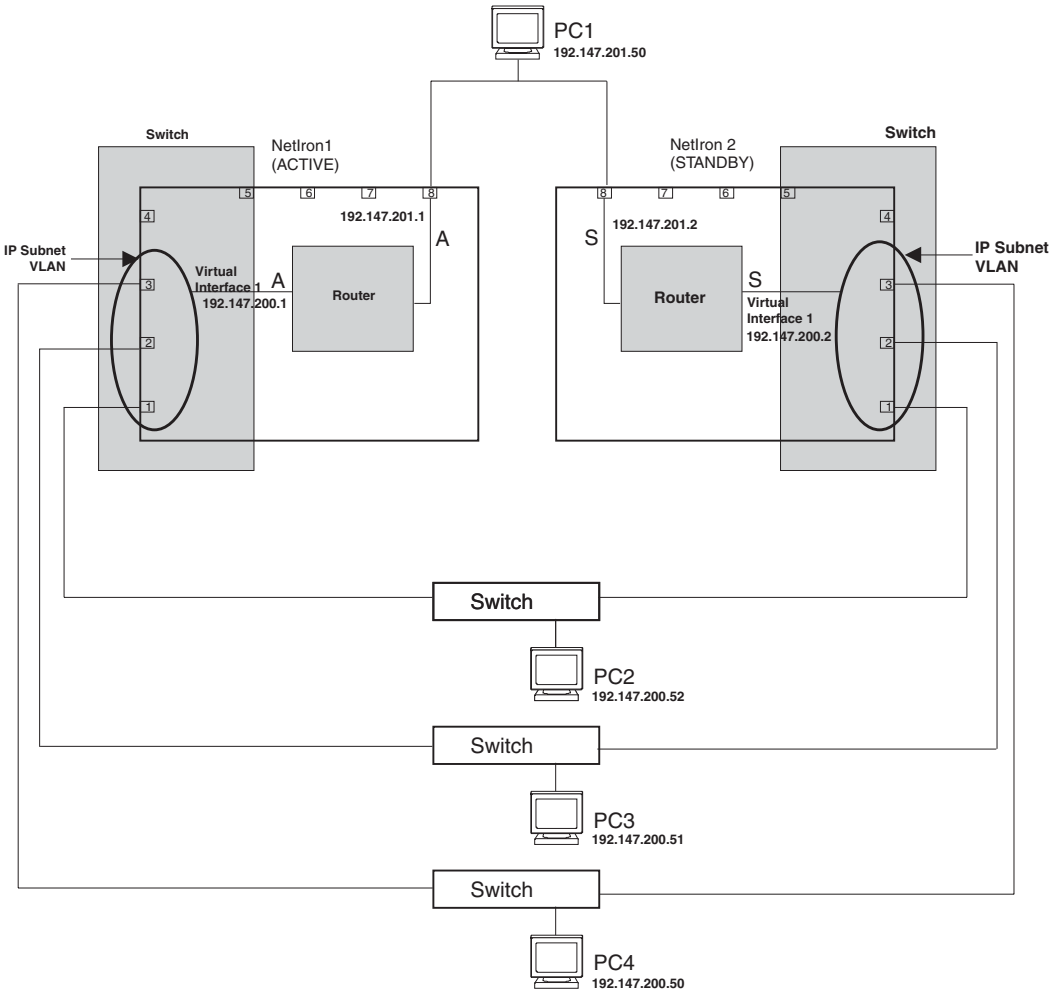
NOTE: If you are configuring a Chassis device, the track port options are listed in a slot/port combination (for example, 1/1, which indicates <slot>/<port>), indicating that the port is resident on a module in slot 1 of the device.

11. Repeat the steps above for each interface that is to be a redundant link. In this example, you would also need to configure interface B for router 1 and interfaces C and D for router 2.
12. Click the Add button to apply the changes to the device's running-config file.
13. Select the Save link at the bottom of the dialog, then select Yes when prompted to save the configuration change to the startup-config file on the device's flash memory.

Configuring FSRP on Virtual Interfaces

A virtual interface will by default remain active until all underlying links go down. If you want the virtual link to go to FSRP standby state when a subset of the ports goes down, you need to configure track ports.

Figure 21.6 Configuring FSRP on virtual interfaces



Configuring Multiple Track Ports for Virtual Interfaces

In Figure 21.6, NetIron1 is the active router and NetIron2 the standby router for all active FSRP interfaces. Suppose you want NetIron1 to go into the FSRP standby state and establish NetIron2 as the active router in case ports 1, 2, 3, or 8 on NetIron1 go down. To do so, you would configure track ports for ports 1, 2, 3, and 8 on NetIron1.

In preparation for track port configuration on NetIron1, you would do the following:

1. Configure an IP subnet VLAN with port membership of 1, 2, and 3 on NetIron1.
2. Enable FSRP on virtual interface 1.
3. Assign an IP address to virtual interface 1.
4. Assign ports 1, 2, 3, and 8 as track ports for virtual interface 1.
5. Assign an IP address to interface 8.
6. Assign ports 1, 2, and 3 as track ports for interface 8.

USING THE CLI

To configure the IP subnet VLAN with port membership of 1, 2, and 3, you would enter the following commands:

```
BigIron(config)# vlan 1
BigIron(config-vlan-1)# ip-subnet 192.147.200.0 255.255.255.0
BigIron(config-vlan-ip-subnet)# static e1 to 3
BigIron(config-vlan-ip-subnet)# router-int ve1
```

To enable FSRP on virtual interface 1 and to configure ports 1, 2, 3, and 8 as its track ports, you would enter the following commands:

```
BigIron(config)# int ve1
BigIron(config-vif-1)# ip address 192.147.200.1 255.255.255.0
BigIron(config-vif-1)# ip fsrp address 192.147.200.1 vir-rtr 192.147.200.100 other-
rtr 192.147.200.2
BigIron(config-vif-1)# ip fsrp addr 192.147.200.1 track port 1 2 3 8
```

To enable FSRP on physical interface 8 and to configure ports 1, 2, and 3 as its track ports, you would enter the following commands:

```
BigIron(config)# int e8
BigIron(config-if-8)# ip address 192.147.201.1 255.255.255.0
BigIron(config-if-8)# ip fsrp address 192.147.201.1 vir-rtr 192.147.201.100 other-
rtr 192.147.200.2
BigIron(config-if-8)# ip fsrp addr 192.147.201.1 track port 1 2 3
BigIron(config-if-8)# end
BigIron# write memory
```

NOTE: After configuring track ports for NetIron1, you would configure NetIron2 similarly. This reciprocal configuration ensures that if NetIron2 becomes the active router, it has track ports that support transfer to a FSRP standby state.

NOTE: Virtual interfaces cannot be assigned as track ports.

USING THE WEB MANAGEMENT INTERFACE

You can select multiple track ports for FSRP on the FSRP configuration sheet.

Chapter 22
Configuring IPX

This chapter describes how to configure the IPX protocol on the Foundry Layer 3 Switches using the CLI and Web management interface.

For information about verifying the connectivity between two devices running IPX, see “Verifying Connectivity” on page 22-16.

To display IPX configuration information and statistics, see “Displaying IPX Configuration Information and Statistics” on page 22-17.

For complete syntax information for the CLI commands shown in this chapter, see the *Foundry Switch and Router Command Line Interface Reference*.

Overview of IPX

The Internet Packet Exchange (IPX) protocol was created by Novell™. IPX is built upon a client-server networking architecture.

The Routing Information Protocol (RIP) and the Service Advertisement Protocol (SAP) are two key components of Novell NetWare and its IPX protocol suite. By default, Novell NetWare versions 3.x and 4.x broadcast RIP and SAP updates at 60 second intervals. NetWare uses these broadcasts to collect information for the routing and service tables that it uses for communicating.

NOTE: IPX/RIP is different from IP/RIP. IP/RIP configuration parameters do not apply to IPX/RIP and IPX/RIP parameters do not apply to IP/RIP.

Multiple IPX Frame Type Support per Interface

Up to four different IPX network numbers and frame encapsulation types can be defined for each IPX interface on a Foundry router. The multiple encapsulation support allows you to define and receive traffic from four separate IPX networks on a single interface. Each network must have a distinct network number and encapsulation type (Ethernet SNAP, Ethernet 802.2, Ethernet 802.3, or Ethernet II).

Configuring IPX

To use IPX on the router, perform the following tasks:

1. Enable IPX on the router.
2. Enable NetBIOS on the system level.

3. Define the network number and frame type, and enable NetBIOS on IPX interfaces (optional).
4. Modify maximum number of RIP and SAP filters supported.
5. Define RIP, SAP, and forward filters (optional).
6. Assign RIP, SAP, and Forward filter groups (optional).
7. Modify the maximum number of SAP and RIP Route entries supported (optional).
8. Modify the hop count increment for RIP and SAP broadcast packets (optional).
9. Modify the maximum advertisement packet size for RIP and SAP packets (optional).
10. Modify the advertisement interval for RIP and SAP updates (optional).
11. Modify the age timer for learned RIP and SAP entries (optional).

Dynamic IPX Configuration

The IPX Protocol is by default disabled at system startup. When you first enable IPX, you must reset the system. However, after you reset the system all changes to the following parameters become effective immediately.

Global Parameters

- Enabling of NetBIOS Allow
- Defining IPX filters—Forward, RIP, and SAP

Interface Parameters

- Adding, deleting, or modifying IPX network numbers and frame types
- Adding, deleting, or modifying filter groups assigned to interfaces
- Modifying the RIP advertisement packet size
- Modifying the SAP advertisement packet size
- Modifying the RIP advertisement interval
- Modifying the SAP advertisement interval
- Modifying the age timer for learned IPX routes
- Modifying the age timer for learned SAP entries

Enable IPX

The IPX Protocol is by default disabled at system startup.

NOTE: Make sure you restart the system after enabling IPX. After you restart, additional IPX parameter settings take effect immediately.

USING THE CLI

To enable IPX, enter the following commands:

```
BigIron(config)# router ipx
BigIron(config)# exit
BigIron# write memory
BigIron# reload
```

Syntax: router ipx

USING THE WEB MANAGEMENT INTERFACE

To enable IPX:

1. Log on to the device using a valid user name and password for read-write access. The System configuration dialog is displayed.

2. Select the Enable radio button next to IPX.
3. Click the Apply button to apply the changes to the device's running-config file.
4. 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.
5. Click on the plus sign next to Command in the tree view to list the command options.
6. Select the Reload link and select Yes when prompted to reload the software. You must reload after enabling IPX to place the change into effect.

Enable NetBIOS

The router can support routing of NetBIOS broadcasts (type 20) over IPX. IPX must be enabled on the router and the interface level for it to be operational. By default, this feature is disabled.

USING THE CLI

To enable NetBIOS on the router (system level), enter the following command:

```
BigIron(config)# ipx netbios-allow
```

Syntax: ipx netbios-allow | netbios-disallow

USING THE WEB MANAGEMENT INTERFACE

To enable NetBIOS (type 20) on the router and an interface:

1. Log on to the device using a valid user name and password for read-write access. The System configuration dialog is displayed.
2. Click on the plus sign next to Configure in the tree view to expand the list of configuration options.
3. Click on the plus sign next to IPX in the tree view to expand the list of IPX option links.
4. Click on the Allow NetBIOS (Type 20) link to display the NetBIOS panel.
5. Select Enable.
6. Click the Apply button to apply the changes to the device's running-config file.
7. 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.

NOTE: After enabling NetBIOS at the global level, you need to enable NetBIOS at the interface level.

Assign IPX Network Number, Frame Type, Enable NetBios on an Interface

Once you enable IPX on the router, you can assign IPX network numbers on an interface-by-interface basis. You also can enable NetBIOS broadcasts on an interface.

USING THE CLI

EXAMPLE:

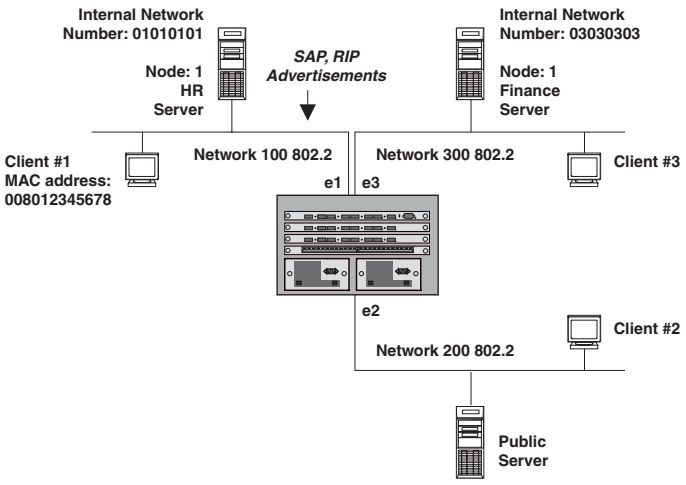
To configure interfaces 1, 2, and 3 with the IPX network number and frame type shown in Figure 22.1, enter the following commands:

```
BigIron(config)# int e 1/1
BigIron(config-if-1/1)# ipx network 100 ethernet_802.2
BigIron(config-if-1/1)# int e 1/2
BigIron(config-if-1/2)# ipx network 200 ethernet_802.2
BigIron(config-if-1/2)# int e 1/3
BigIron(config-if-1/3)# ipx network 300 ethernet_802.2
```

Syntax: ipx network <network-number> <frame-type> [netbios-allow | netbios-disallow]

NOTE: Once you configure an interface with a network number and frame type, you can define filters and assign them to the interface.

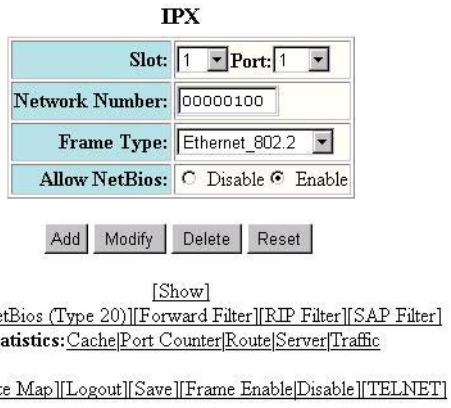
Figure 22.1 Defining and assigning IPX Forward, RIP and SAP filters



USING THE WEB MANAGEMENT INTERFACE

To assign IPX to interfaces 1, 2 and 3 as shown in Figure 22.1:

1. Log on to the device using a valid user name and password for read-write access. The System configuration dialog is displayed.
2. Click on the plus sign next to Configure in the tree view to expand the list of configuration options.
3. Click on the plus sign next to IPX in the tree view to expand the list of IPX option links.
4. Click on the Interface link.
   - If the device does not have an IPX interface configured, the IPX configuration panel is displayed, as shown in the following example.
   - If an IPX interface is already configured and you are adding a new one, click on the Configure IPX Interface link to display the IPX interface configuration panel, as shown in the following example.
   - If you are modifying an existing IPX interface, click on the Modify button to the right of the row describing the interface to display the IPX configuration panel, as shown in the following example.



5. Select the port or slot/port numbers to be configured as an IPX interface from the pull down menu.
6. Enter the network number.
7. Select the frame type from the pull down menu.
8. Enable NetBIOS if desired.
9. Click the Add button to apply the changes to the device's running-config file.
10. 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.

Define and Assign a Forward Filter and Group

You can define a forward filter to allow a remote IPX client access to a restricted-access server. You can define up to 32 forward filters on a router. Once you define the filter, you assign the filter to an interface by placing the filter in a forward filter group.

NOTE: A network number and frame type must be defined for the IPX interface before defining a forward filter.

EXAMPLE:

To allow IPX Client 1 on network 100 access to the finance server in Network 300 (Figure 22.1), define the following forward filter at the Global Level and then assign the filter to port 3 as a filter group.

NOTE: You can assign forward filters to either the input or output traffic on an interface.

USING THE CLI

```
BigIron(config)# ipx forward-filter 1 permit 100 008012345678 03030303 1 451
BigIron(config)# int e 1/3
BigIron(config-if-1/3)# ipx forward-filter-group in 1
```

Syntax: ipx forward-filter <filter-id> permit | deny <source-network-number> | any <source-node-number> | any <destination-network-number> | any <destination-node-number> | any <destination-socket-number> | any

Syntax: ipx forward-filter-group in | out <filter-id>

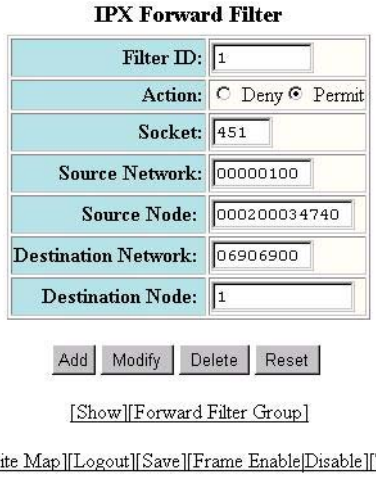
NOTE: When you define filters, the network number for a server is its internal network number. The node number for a client is the client's MAC address. The value 1 represents a server.

USING THE WEB MANAGEMENT INTERFACE

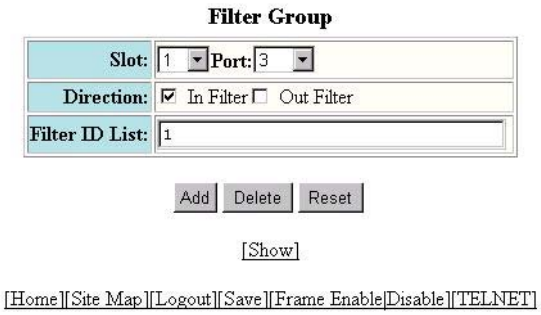
EXAMPLE:

To allow IPX Client 1 on network 100 access to the finance server in Network 300 (Figure 22.1), define the following forward filter at the Global Level and then assign it to port 3 as a filter group.

1. Log on to the device using a valid user name and password for read-write access. The System configuration dialog is displayed.
2. Click on the plus sign next to Configure in the tree view to expand the list of configuration options.
3. Click on the plus sign next to IPX in the tree view to expand the list of IPX option links.
4. Click on the Forward Filter link.
   - If the device does not have an IPX forward filter configured, the IPX Forward Filter configuration panel is displayed, as shown in the following example.
   - If an IPX forward filter is already configured and you are adding a new one, click on the Add Forward Filter link to display the IPX Forward Filter configuration panel, as shown in the following example.
   - If you are modifying an existing IPX forward filter, click on the Modify button to the right of the row describing the filter to display the IPX Forward Filter configuration panel, as shown in the following example.



5. Enter a filter ID value from 1 – 32.
6. Select either Permit or Deny.
7. Enter the appropriate number for the destination socket of the application running in the Socket field. If you enter all zeros in this field, the filter will accept any socket.
8. Enter the Source Network Address on which you want to filter traffic. If you enter all zeros in this field, the filter will accept any source network.
9. Enter the address of the Source Node within the source network on which you want to filter traffic.
10. Enter the Destination network number. If you enter all zeros in this field, the filter will accept any destination network number.
11. Enter the Destination Node network number. If you enter all zeros in this field, the filter will accept any destination node network number.
12. Click the Add button to apply the changes to the device's running-config file.
13. Select the Forward Filter Group link.
    - If the device does not have an IPX forward filter group configured, the Filter Group configuration panel is displayed, as shown in the following example.
    - If an IPX forward filter group is already configured and you are adding a new one, click on the Add Forward Filter Group link to display the IPX Filter Group configuration panel, as shown in the following example.
    - If you are modifying an existing IPX forward filter group, click on the Modify button to the right of the row describing the group to display the IPX Filter Group configuration panel, as shown in the following example.



14. Select the port or slot/port combination to which you are assigning the filter(s).
15. Check either or both of the In Filter and Out Filter boxes. If you check the In Filter box, all incoming traffic is filtered as defined. If you check the Out Filter box, all outgoing traffic is filtered. By selecting both the In Filter and Out Filter boxes, you can assign the filters to both incoming and outgoing traffic.
16. Enter the filter ID(s) that you want to assign to the port. You can enter multiple filters entries separated by commas or blanks.
17. Click the Add button to apply the changes to the device's running-config file.
18. 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.

Define and Assign an IPX/RIP Filter and Group

You can define a filter for a router to block RIP routes being advertised to other parts of the network. You define RIP filters at the global level and assign them on either a global or interface basis. You can apply filters to either incoming or outgoing traffic. You can define up to 128 IPX/RIP filters on a router.

NOTE: An IPX interface must be defined on the router before you can assign a filter to that interface.

EXAMPLE:

To block RIP routes from being advertised outside of Network 100, shown in Figure 22.1, define and assign the following RIP filter on interface 1.

USING THE CLI

```
BigIron(config)# ipx rip-filter 1 deny 100 01010101 any
BigIron(config)# int e 1/1
BigIron(config-if-1/1)# ipx rip-filter-group in 1
```

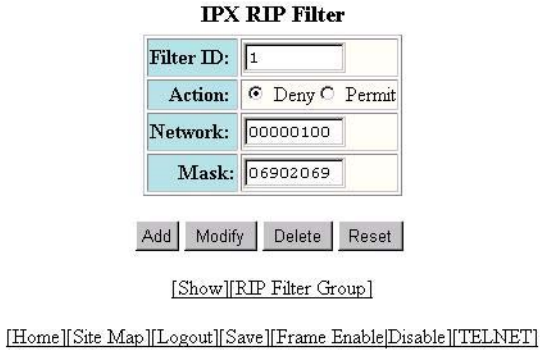
Syntax: ipx rip-filter <filter-id> permit | deny <network-number> | any <network-mask> | any

Syntax: ipx rip-filter-group in | out <filter-id>

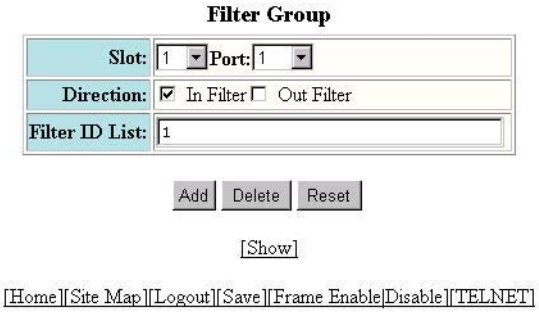
USING THE WEB MANAGEMENT INTERFACE

1. Log on to the device using a valid user name and password for read-write access. The System configuration dialog is displayed.
2. Click on the plus sign next to Configure in the tree view to expand the list of configuration options.
3. Click on the plus sign next to IPX in the tree view to expand the list of IPX option links.
4. Click on the RIP Filter link.
   - If the device does not have an IPX RIP filter configured, the IPX RIP Filter configuration panel is displayed, as shown in the following example.

- If an IPX RIP filter is already configured and you are adding a new one, click on the Add RIP Filter link to display the IPX RIP Filter configuration panel, as shown in the following example.
- If you are modifying an existing IPX RIP filter, click on the Modify button to the right of the row describing the filter to display the IPX RIP Filter configuration panel, as shown in the following example.



5. Enter an filter ID in the Filter ID field.
6. Select either Permit or Deny.
7. Enter the source network address on which you want to filter traffic in the Network field. You also can assign a wildcard value of all zeros (00000000) to allow all entries. The zeroes appear as 'any' in the display.
8. Enter the source network address mask for the network address in the Mask field. You can assign a wildcard value of all zeros (00000000) to allow all entries. The zeroes appear as 'any' in the display.
9. Click the Add button to apply the changes to the device's running-config file.
10. Select the RIP Filter Group link.
    - If the device does not have an IPX RIP filter group configured, the Filter Group configuration panel is displayed, as shown in the following example.
    - If an IPX RIP filter group is already configured and you are adding a new one, click on the Add RIP Filter Group link to display the Filter Group configuration panel, as shown in the following example.
    - If you are modifying an existing IPX RIP filter group, click on the Modify button to the right of the row describing the group to display the Filter Group configuration panel, as shown in the following example.



11. Select the port to which you want to assign the filter(s).
12. Check either or both of the In Filter and Out Filter boxes. If you check the In Filter box, all incoming traffic is filtered. If you check the Out Filter box, all outgoing traffic is filtered. If you check both In Filter and Out Filter, the assigned filters apply to both incoming and outgoing traffic.

13. Enter the filter ID(s) you want to assign to the port. You can enter multiple filter entries separated by commas or blanks.
14. Click the Add button to apply the changes to the device's running-config file.
15. 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.

Configuring IPX SAP Access Control Lists (ACLs)

You can configure Access Control Lists (ACLs) for filtering Service Advertisement Protocol (SAP) replies sent on a Layer 3 Switch's IPX interfaces. You configure IPX SAP access lists on a global basis, then apply them to the IPX inbound or outbound filter group on specific interfaces. You can configure up to 1024 IPX SAP access lists on Chassis devices and up to 32 on Stackable devices. The default number of IPX SAP access lists is set at 256 globally on Chassis devices. On stackable devices the default and maximum number of IPX SAP access lists is 32. Up to 32 access lists can be applied per interface. The same access list can be applied to multiple interfaces.

When you configure more than one access list on an IPX interface, the software applies the access lists in numerical order. For example, if you configure access lists 1, 10, and 32 and apply them to an interface, the software applies access list 1 first, then access list 10, then access list 32. This is true regardless of the order in which you configure the access lists. At the first match, the software takes the action specified by the access list (deny or permit) and stops comparing the update against the access lists.

IPX SAP access lists apply to SAP updates sent or received by the Layer 3 Switch. You can apply them to a port's inbound or outbound IPX traffic.

NOTE: IPX access lists replace the IPX filter mechanism in software releases earlier than 06.0.00. The older commands are supported for backward compatibility but are not listed in the on-line help. If the devices' startup-config file contains IPX filter commands of the older format, they are replaced by equivalent IPX ACL commands when you save the device's configuration while running 06.0.00 or later.

Before you configure an access list on an IPX interface, all SAP updates are sent and received by default. However, once you configure an access filter, the default action changes from permit to deny. Thus, SAP updates that are not explicitly permitted are denied. To change the default action to permit, configure SAP access list 32 to permit all updates on all networks.

NOTE: Each IPX SAP access list is a single filter. This is different from the system-wide ACLs, which each can contain multiple individual filters. See "Access Control List" on page 6-1.

To configure IPX access lists, use the following CLI method.

USING THE CLI

To configure three IPX access lists and apply them to IPX interfaces on port 1/1, enter the following commands:

```
BigIron(config)# router ipx
BigIron(config)# ipx sap-access-list 1 deny abcd
BigIron(config)# ipx sap-access-list 10 deny efef.1234.1234.1234
BigIron(config)# ipx sap-access-list 32 permit -1 0
BigIron(config)# exit
BigIron(config)# int e 1/1
BigIron(config-if-1/1)# ipx sap-filter-group out 1 10 32
BigIron(config-if-1/1)# write memory
```

In this example, access list 1 denies all SAP updates containing IPX network abcd. Access list 10 denies SAP updates for print server "Prt1" from network efef, node 1234.1234.1234. Access list 32 ensures that all updates that are not denied by the preceding access lists are permitted.

Syntax: [no] ipx sap-access-list <num> deny | permit <network>[.<node>] [<network-mask>.<node-mask>] [<service-type> [<server-name>]]

Syntax: [no] ipx sap-filter-group in | out <num> [<num>…]

The <num> parameter specifies the access list number and can be from 1 – 32.

The **deny | permit** parameter specifies whether the Layer 3 Switch allows the SAP update or denies it.

The <network>[.<node>] parameter specifies the IPX network. Optionally, you also can specify a specific node (host) on the network. The <network> parameter can be an eight-digit hexadecimal number from 1 – FFFFFFFE. To specify all networks ("any"), enter –1 as the network number. If the network number has leading zeros, you do not need to specify them. For example, you can specify network 0000abab as "abab".

The node is a 48-bit value represented by three four-digit numbers joined by periods; for example, 1234.1234.1234.

The [<network-mask>.<node-mask>] parameter lets you specify a comparison mask for the network and node. The mask consists of zeros (0) and ones (f). Ones indicate significant bits. For example, to configure a mask that matches on network abcdef*xx*, where *xx* can be any value and the node address can be any value, specify the following mask: ffffff00.0000.0000.0000

NOTE: To apply an ACL for filtering GNS replies to an interface, you must use the **ipx output-gns-filter** command instead of the **ipx sap-filter-group** command. See "Filter GNS Replies" on page 22-10.

The **in | out** parameter of the **ipx sap-filter-group** command specifies whether the ACLs apply to incoming traffic or outgoing traffic.

USING THE WEB MANAGEMENT INTERFACE

You cannot configure a SAP access list using the Web management interface.

Enable Round-Robin GNS Replies

By default, the Layer 3 Switch replies to a GNS request with the most recently learned server supporting the requested service. You configure the Layer 3 Switch to instead use round-robin to rotate among servers of a given service type when responding to GNS requests. To do so, use one of the following methods.

USING THE CLI

To enable the Layer 3 Switch to use round-robin to select servers for replies to GSN requests, enter the following commands:

```
BigIron(config)# ipx gns-round-robin
BigIron(config)# write memory
```

Syntax: [no] ipx gns-round-robin

USING THE WEB MANAGEMENT INTERFACE

You cannot enable round-robin for GNS replies using the Web management interface.

Filter GNS Replies

You can use IPX access lists to permit or deny specific services and servers in GNS replies to specific IPX nodes (hosts). To do so, use either of the following methods to configure IPX access lists that include service and server information, then apply them to specific ports.

USING THE CLI

To configure IPX ACLs and apply them to a port to control responses to GNS requests on that port, enter commands such as the following:

```
BigIron(config)# router ipx
BigIron(config-ipx-router)# ipx sap-access-list 2 deny efff 47 Prt0
BigIron(config-ipx-router)# ipx sap-access-list 20 deny aaaa.bbbb.cccc.dddd 47 Prt1
BigIron(config-ipx-router)# ipx sap-access-list 32 permit -1 0
BigIron(config-ipx-router)# exit
BigIron(config)# int e 1/1
BigIron(config-if-1/1)# ipx output-gns-filter 10 20 32
BigIron(config-if-1/1)# write memory
```


The commands in this example configure three ACLs. Two of the ACLs contain server network, service type, and server information and deny reporting these servers to the clients. For example, ACL 2 does not permit the Layer 3 Switch from sending server "Prt0" with network efff in GNS replies to the client.

ACL 32 changes the default action from deny to permit. All GNS replies that are not explicitly denied by other ACLs are permitted by this one.

You can use IPX SAP access lists with filter IDs from 1 – 32 for GNS reply filters.

Syntax: [no] ipx sap-access-list <num> deny | permit <network>[.<node>] [<network-mask>.<node-mask>] [<service-type> [<server-name>]]

The <service-type> [<server-name>] parameter lets you specify a service type and, optionally, a specific server. Use these parameters when you are configuring an ACL for filtering Get Nearest Server (GNS) replies. The service type is a hexadecimal number. To specify all service types ("any"), enter 0. For a list of service types, see the software documentation for your IPX servers. If you also enter the server name, the access list applies only to updates for that server, not to other serves of the same type.

For information about the other parameters, see "Configuring IPX SAP Access Control Lists (ACLs)" on page 22-9.

Syntax: [no] ipx output-gns-filter <num> [<num>…]

USING THE WEB MANAGEMENT INTERFACE

You cannot configure GNS reply filters using the Web management interface.

Disable GNS Replies

When IPX is enabled in the Layer 3 Switch, the device responds to all GNS requests by default. You can disable GNS replies on individual Layer 3 Switch ports. Use one of the following methods to do so.

USING THE CLI

To disable IPX GNS replies on port 1/1, enter the following commands. GNS replies are disabled for all IPX interfaces on the port.

```
BigIron(config)# int eth 1/1
BigIron(config-if-1/1)# ipx gns-reply-disable
BigIron(config-if-1/1)# write memory
```

Syntax: [no] ipx gns-reply-disable

USING THE WEB MANAGEMENT INTERFACE

You cannot disable IPX GNS replies using the Web management interface.

Modify Maximum SAP and RIP Route Entries

You can define the maximum number of IPX/RIP and IPX/SAP routes that the router can store and forward.

- From 64 – 8192 RIP entries can be defined on a Stackable device or Chassis device. The default number of RIP entries supported is 2048.
- From 64 – 8192 SAP entries can be defined on a Stackable device or Chassis device. The default number of SAP entries supported is 4096.

NOTE: IPX must be enabled on the router for these items to be configurable.

USING THE CLI

To limit the number of RIP entries stored to 3500 from a default of 2048, enter the following command:

```
BigIron(config)# system-max ipx-rip-entry 3500
```

Syntax: system-max ipx-rip-entry <value>

To limit the number of SAP entries stored to 6000 from a default of 4096, enter the following command:

```
BigIron(config)# system-max ipx-sap-entry 6000
```

Syntax: system-max ipx-sap-entry <value>

USING THE WEB MANAGEMENT INTERFACE

To modify the maximum number of RIP or SAP route entries supported on a router:

1. Log on to the device using a valid user name and password for read-write access. The System configuration panel is displayed.
2. Select the Max-Parameter link to display the Configure System Parameter Maximum Value table. This table lists the settings and valid ranges for all the configurable table sizes on the device.
3. Click the Modify button next to ipx-rip-entry or ipx-sap-entry.
4. Enter the new value for the table size. The value you enter specifies the maximum number of entries the table can hold.
5. Click Apply to save the changes to the device's running-config file.
6. 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.
7. Click on the plus sign next to Command in the tree view to list the command options.
8. Select the Reload link and select Yes when the Web management interface asks you whether you really want to reload the software. Changes to table sizes do not take effect until you reload the software.

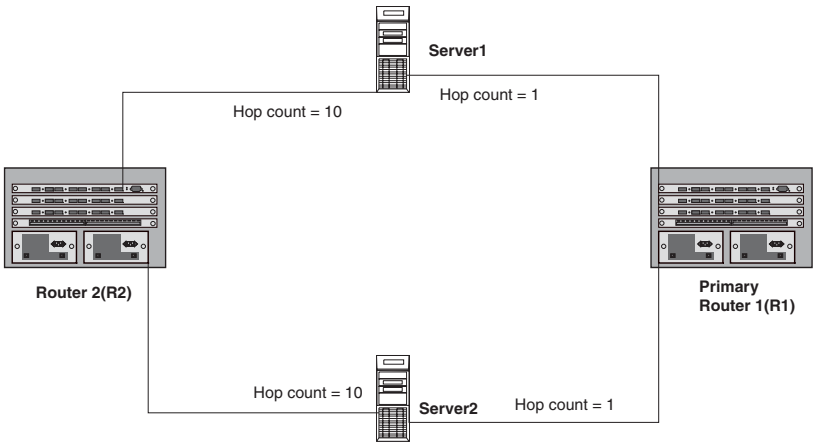
Modify RIP and SAP Hop Count Increment

You can modify the incremental value (hop) that the router adds to a RIP or SAP record before propagating the record to the next interface. By default, a value of one is added to a record before it is broadcast to the next interface.

In a network of parallel routers, the router that receives a RIP or SAP record with the lowest hop count is seen as the router with the most optimal information and is seen as the primary router. As primary router, it is elected to forward the packet to the next interface.

You can manage which router is selected as the primary router by a host by modifying the hop count assigned to an IPX interface. For example, in Figure 22.2, an administrator wants to ensure that all traffic between server1 and server2 is routed through router 1 and that router 1 is seen as the primary router. To ensure that this occurs, the administrator can assign higher hop counts (for example, 10) to the router interfaces on router 2.

Figure 22.2 Using higher hop count assignments to bias traffic away from the router



USING THE CLI

To increase the hop count increment assessed to interface 5, enter the following commands:

```
BigIron(config)# int e 1/5
BigIron(config-if-1/5)# ipx-rip-update-hop-count-increment 10
BigIron(config-if-1/5)# ipx-sap-update-hop-count-increment 10
```

Syntax: ipx-rip-update-hop-count-increment <2-15>, ipx-sap-update-hop-count-increment <2-15>

USING THE WEB MANAGEMENT INTERFACE

You cannot modify hop count increments using the Web management interface.

Modify the RIP Advertisement Packet Size

The default IPX RIP packet size is 432 bytes, which allows 50 routes plus 32 bytes of header in an IPX RIP update packet. Each route requires eight bytes. You can configure the packet size to be from 40 bytes (enough for one route) – 1488 bytes (enough for 182 routes).

NOTE: You can specify packet length that does not fall evenly on a route or server boundary. The device will use the packet size but will include only the number of routes or servers that fit entirely within the packet.

To change the RIP advertisement packet size, use the following CLI method.

USING THE CLI

EXAMPLE:

To change the maximum packet size of IPX RIP advertisements sent on interface 1/1 from the default 432 bytes to 832 bytes, enter the following command. This command increases the number of IPX RIP routes an advertisement packet holds from 50 to 100.

```
BigIron(config) int e 1/1
BigIron(config-if-1/1) ipx rip-max-packetsize 832
BigIron(config-if-1/1) write memory
```

Syntax: ipx rip-max-packetsize <bytes>

The number of bytes can be from 40 bytes (enough for one route) – 1488 bytes (enough for 182 routes). The default is 432 bytes.

USING THE WEB MANAGEMENT INTERFACE

You cannot modify the RIP advertisement packet size using the Web management interface.

Modify the SAP Advertisement Packet Size

The default IPX SAP packet size is 480 bytes, which allows seven servers plus 32 bytes of header in an IPX SAP update packet. Each server requires 64 bytes. You can configure the packet size to be from 96 bytes (enough for one server) – 1440 bytes (enough for 22 servers).

NOTE: You can specify packet length that does not fall evenly on a route or server boundary. The device will use the packet size but will include only the number of routes or servers that fit entirely within the packet.

To change the SAP advertisement packet size, use the following CLI method.

USING THE CLI

EXAMPLE:

To change the maximum number of bytes in IPX SAP advertisements sent on interface 5/1 from 480 to 672 (enough for 10 servers plus the 32 bytes of packet header), enter the following commands:

```
BigIron(config) int e 5/1
BigIron(config-if-5/1) ipx sap-max-packetsize 672
BigIron(config-if-5/1) write memory
```


Syntax: ipx sap-max-packetsize <bytes>

The number of bytes can be from 96 bytes (enough for one server) – 1440 bytes (enough for 22 servers). The default is 480 bytes.

USING THE WEB MANAGEMENT INTERFACE

You cannot modify the SAP advertisement packet size using the Web management interface.

Modify the RIP Advertisement Interval

The IPX RIP advertisement interval specifies how often the Layer 3 Switch sends IPX RIP updates to neighboring IPX routers. The update intervals are separate for RIP and SAP and are configurable on an individual interface basis.

By default, the Layer 3 Switch sends an IPX RIP update every 60 seconds. You can change the interval to be from 10 – 65535 seconds. You cannot disable the advertisements.

NOTE: If you change an advertisement interval, you do not need to change the age time. The software automatically calculates the age time by multiplying the advertisement interval times the age timer, which is 3 by default.

To change the RIP advertisement interval, use the following CLI method.

USING THE CLI

EXAMPLE:

To change the advertisement interval for IPX RIP advertisements sent on interface 1/1 from 60 seconds to 30 seconds, enter the following commands:

```
BigIron(config) int e 1/1
BigIron(config-if-1/1) ipx update-time 30
BigIron(config-if-1/1) write memory
```

Syntax: ipx update-time <interval>

The <interval> can be from 10 – 65535 seconds. The default is 60.

USING THE WEB MANAGEMENT INTERFACE

You cannot modify the RIP advertisement interval using the Web management interface.

Modify the SAP Advertisement Interval

The IPX SAP advertisement interval specifies how often the Layer 3 Switch sends IPX SAP updates to neighboring IPX routers. The update intervals are separate for RIP and SAP and are configurable on an individual interface basis.

By default, the Layer 3 Switch sends an IPX SAP update every 60 seconds. You can change the interval to be from 10 – 65535 seconds. You cannot disable the advertisements.

NOTE: If you change an advertisement interval, you do not need to change the age time. The software automatically calculates the age time by multiplying the advertisement interval times the age timer, which is 3 by default.

To change the SAP advertisement packet size, use the following CLI method.

USING THE CLI

EXAMPLE:

To change the advertisement interval for IPX SAP advertisements sent on interface 1/1 from 60 seconds to 120 seconds, enter the following commands:

```
BigIron(config) int e 1/1
BigIron(config-if-1/1) ipx sap-interval 120
BigIron(config-if-1/1) write memory
```

Syntax: ipx sap-interval <interval>

The <interval> can be from 10 – 65535 seconds. The default is 60.

USING THE WEB MANAGEMENT INTERFACE

You cannot modify the SAP advertisement interval using the Web management interface.

Modify the Age Timer for Learned IPX Routes

The age timer specifies how many seconds a learned IPX route can remain in the Layer 3 Switch's IPX route table before aging out.

The software calculates the age time by multiplying the advertisement interval times the age timer. For example, the default age time for IPX routes is 180 seconds, which is 60 (the default advertisement interval) multiplied by 3 (the default age timer).

You can configure the age timer for RIP to a value from 1 – 65535. The default is 3. You cannot disable the age timer.

To change the age timer for learned IPX routes, use the following CLI method.

USING THE CLI

To change the age timer for IPX routes from 3 to 4 on interface 1/1, enter the following commands.

```
BigIron(config) int e 1/1
BigIron(config-if-1/1) ipx rip-multiplier 4
BigIron(config-if-1/1) write memory
```

Syntax: ipx rip-multiplier <num>

The <num> parameter specifies the age time and can be from 1 – 65535. The default is 3.

USING THE WEB MANAGEMENT INTERFACE

You cannot modify the route age timer using the Web management interface.

Modify the Age Timer for Learned SAP Entries

The age timer specifies how many seconds a learned IPX server can remain in the Layer 3 Switch's IPX service table before aging out.

The software calculates the age time by multiplying the advertisement interval times the age timer. For example, the default age time for IPX service entries is 180 seconds, which is 60 (the default advertisement interval) multiplied by 3 (the default age timer).

You can configure the age timer for SAP to a value from 1 – 65535. The default is 3. You cannot disable the age timer.

To change the age timer for learned SAP entries, use the following CLI method.

USING THE CLI

To change the age timer for IPX servers from 3 to 2 on interface 5/1, enter the following commands.

```
BigIron(config) int e 5/1
BigIron(config-if-5/1) ipx sap-multiplier 2
BigIron(config-if-5/1) write memory
```


Syntax: ipx sap-multiplier <num>

The <num> parameter specifies the age time and can be from 1 – 65535. The default is 3.

USING THE WEB MANAGEMENT INTERFACE

You cannot modify the SAP age timer using the Web management interface.

Verifying Connectivity

Foundry Layer 3 Switches support IPX ping, which enables you to verify connectivity between the Layer 3 Switch and a target device that also supports IPX ping.

For example, to initiate the Layer 3 Switch to send 100000 pings to a target device with the IPX network number of A5001234 and node number of 00e0.52ab.4921, enter the following command at the User EXEC or Privileged EXEC level of the CLI:

```
BigIron# ipx-ping a5001234 00e0.52ab.4921 count 100000
```

Syntax: ipx-ping <network-number> <node-number> [count <pings>] [timeout <milliseconds>] [ttl <number>] [verify] [quiet] [data <1-to-4 byte hex>] [size <byte>] [brief [max-print-per-sec <number>]]

The <network-number> parameter indicates the target device's assigned 4-byte external/internal network number.

The <node-number> parameter indicates the target device's assigned 6-byte node number. For a client, the node number is usually the client's MAC address. For a server, the node number is usually 0.0.1.

The optional **count** <pings> parameter indicates the number of pings the Layer 3 Switch sends to the target. You can specify from 1 – 4294967296 pings.The default is 1 ping.

The optional **timeout** <milliseconds> parameter specifies how many milliseconds the Layer 3 Switch waits for a reply from the pinged device. You can specify a timeout from 1 – 4294967296 milliseconds. The default is 5000 (5 seconds).

The optional **ttl** <number> parameter specifies the maximum number of hops to a target device as determined by IPX's transport control feature. You can specify a transport control value from 0 – 16. The default is 1.

The optional **verify** parameter 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 optional **quiet** parameter 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 optional **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 IPX ping message (payload) portion of the packet.

The optional **size** <byte> parameter specifies the size of the IPX ping data portion of the packet. This is the payload and does not include the header. You can specify from 0 – 500. The default is 12.

NOTE: For numeric parameter values, 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 optional **brief** parameter causes the Layer 3 Switch to display ping test characters. The Layer 3 Switch supports the following ping test characters:

- ! – 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.

The optional **max-print-per-sec** <number> parameter specifies the maximum number of target responses the Layer 3 Switch can display per second while in brief mode. You can specify from 0 – 2047. The default is 2047.

Displaying IPX Configuration Information and Statistics

You can use CLI commands and Web management options to display the following IPX information:

- Global IPX parameter settings – see “Displaying Global IPX Configuration Information” on page 22-17.
- IPX interfaces – see “Displaying IPX Interface Information” on page 22-19.
- IPX forwarding cache – see “Displaying the IPX Forwarding Cache” on page 22-21.
- IPX route table – see “Displaying the IPX Route Table” on page 22-22.
- IPX server table – see “Displaying the IPX Server Table” on page 22-23.
- IPX traffic statistics – see “Displaying IPX Traffic Statistics” on page 22-24.

Displaying Global IPX Configuration Information

To display global IPX configuration information for the router, use one of the following methods.

USING THE CLI

To display IPX configuration information, enter the following command at any CLI level:

```
BigIron> show ipx

IPX Enabled
NetBIOS (type 20): Disallowed

Maximum RIP entries: 2048
Maximum SAP entries: 4096

Maximum IPX RIP filters: 32
Maximum IPX SAP filters: 32
Maximum IPX forward filters: 32
```

Syntax: show ipx

This display shows the following information.

Table 22.1: CLI Display of Global IPX Configuration Information

This Field...	Displays...
IPX Enabled	Verifies that IPX is enabled. **Note**: If IPX is disabled, the following message is displayed in stead: “ipx not running”
IPX NetBIOS (type 20)	Indicates whether IPX is configured to allow NetBIOS type 20 packets. This field can have one of the following values: • Allowed • Disallowed To change this parameter, see “Enable NetBIOS” on page 22-3.

Table 22.1: CLI Display of Global IPX Configuration Information (Continued)

This Field...	Displays...
Maximum IPX RIP filters	How many IPX route filters you can configure in the router. On some devices, you can change this value by changing the amount of memory allocated for the filters. See the "Displaying and Modifying System Parameter Default Settings" section of the "Configuring Basic Features" chapter of the *Foundry Switch and Router Installation and Basic Configuration Guide*.
Maximum IPX SAP filters	How many IPX service filters you can configure in the router. On some devices, you can change this value by changing the amount of memory allocated for the filters. See the "Displaying and Modifying System Parameter Default Settings" section of the "Configuring Basic Features" chapter of the *Foundry Switch and Router Installation and Basic Configuration Guide*.
Maximum IPX forward filters	How many IPX forward filters you can configure in the router. On some devices, you can change this value by changing the amount of memory allocated for the filters. See the "Displaying and Modifying System Parameter Default Settings" section of the "Configuring Basic Features" chapter of the *Foundry Switch and Router Installation and Basic Configuration Guide*.

USING THE WEB MANAGEMENT INTERFACE

1. Log on to the device using a valid user name and password for read-only or read-write access. The System configuration dialog is displayed.
2. Verify that the Enable option is selected next to IPX. If the option is not selected and you want to enable IPX, see "Enable IPX" on page 22-2.

To determine whether NetBIOS is enabled or disabled:

1. Click on the plus sign next to Configure.
2. Click on the plus sign next to IPX in the tree view to expand the list of IPX option links.
3. Click on the Allow NetBIOS (Type 20) link. Verify that Enable is selected.

To view the maximum number of IPX filters you can configure:

1. Click the Home link from any panel to display the System configuration panel.
2. Select the Max-Parameter link to display the Configure System Parameter Maximum Value table. This table lists the settings and valid ranges for all the configurable table sizes on the device.
3. Scroll down to display the values in the Max Current Value field for the following parameters:
   - ipx-forward-filter – IPX forward filters
   - ipx-rip-filter – IPX RIP filters
   - ipx-sap-filter – IPX SAP filters

Displaying IPX Interface Information

To display IPX interface information for the router, use one of the following methods.

USING THE CLI

To display IPX interface information, enter the following command at any CLI level:

```
BigIron# show ipx interface ethernet 3/5

Interface Ethernet 3/5
  MAC address: 00e0.5284.0b44  Port state: UP
  IPX network:       0000ABCD  Frame type: ethernet_snap   Allow NetBIOS: NO
  rip-interval: 60  rip-max-packet-size: 432  rip-multiplier: 3
  sap-interval: 60  sap-max-packet-size: 480  sap-multiplier: 3
```

Syntax: show ipx interface [ethernet <portnum> | ve <num>]

The **ethernet** <portnum> parameter lets you specify a router port.

The **ve** <num> parameter lets you specify a virtual interface (VE).

This display shows the following information.

Table 22.2: CLI Display of IPX Interface Information

This Field...	Displays...
Interface	The port or virtual interface on which the IPX interface is configured.
MAC address	The MAC address of the interface.
Port state	The state of the interface. The state can be one of the following: • DOWN • UP
IPX network	The IPX network number.
Frame type	The frame type of the network. The frame type can be one of the following: • ethernet_802.2 • ethernet_802.3 • ethernet_ii • ethernet_snap
Allow NetBIOS	Indicates whether the interface allows NetBIOS traffic. This field can have the following values: • NO • YES
rip-interval	The RIP advertisement interval. The RIP advertisement interval specifies how often the Layer 3 Switch sends IPX RIP updates to neighboring IPX routers. To modify this parameter, see “Modify the RIP Advertisement Interval” on page 22-14.

Table 22.2: CLI Display of IPX Interface Information (Continued)

This Field...	Displays...
rip-max-packet-size	The maximum packet size for IPX RIP updates. The default IPX RIP packet size is 432 bytes, which allows 50 routes plus 32 bytes of header in an IPX RIP update packet. To modify this parameter, see “Modify the RIP Advertisement Packet Size” on page 22-13.
rip-multiplier	The age timer for learned IPX routes. The age timer specifies how many seconds a learned IPX route can remain in the Layer 3 Switch's IPX route table before aging out. To modify this parameter, see “Modify the Age Timer for Learned IPX Routes” on page 22-15.
sap-interval	The SAP advertisement interval. The IPX SAP advertisement interval specifies how often the Layer 3 Switch sends IPX SAP updates to neighboring IPX routers. To modify this parameter, see “Modify the SAP Advertisement Interval” on page 22-14.
sap-max-packet-size	The maximum packet size for IPX SAP advertisements. The default IPX SAP packet size is 480 bytes, which allows seven servers plus 32 bytes of header in an IPX SAP update packet. To modify this parameter, see “Modify the SAP Advertisement Packet Size” on page 22-13.
sap-multiplier	The age timer for learned SAP entries. The age timer specifies how many seconds a learned IPX server can remain in the Layer 3 Switch's IPX service table before aging out. To modify this parameter, see “Modify the Age Timer for Learned SAP Entries” on page 22-15.

USING THE WEB MANAGEMENT INTERFACE

To display IPX interface information:

1. Log on to the device using a valid user name and password for read-only or read-write access. The System configuration dialog is displayed.
2. Click on the plus sign next to Configure in the tree view.
3. Click on the plus sign next to IPX in the tree view to expand the list of IPX option links.
4. Click on the Interface link to display the IPX interface table.

Displaying the IPX Forwarding Cache

To display the IPX forwarding cache for the router, use one of the following methods.

USING THE CLI

To display the IPX forwarding cache, enter the following command at any CLI level:

```
BigIron> show ipx cache
Total number of IPX cache entries 3

Forwarding

Index  Network   Router      Out-Filter  Frame-Type    Port
1     11110007  0000.0000.0000  off      ethernet_802.3    7
2     11110005  0000.0000.0000  off      ethernet_802.3    5
3     32D564FA  00a0.24bf.89ca  off      ethernet_802.3    5
```

Syntax: show ipx cache [<num(hex)>]

The <num(hex)> parameter lets you specify an IPX network number.

This display shows the following information.

Table 22.3: CLI Display of IPX Forwarding Cache

This Field...	Displays...
Total number of IPX cache entries	The number of entries in the forwarding cache.
Index	The row number of this entry in the cache.
Network	The network containing the destination node.
Router	The MAC address of the next-hop IPX router. If the destination is local, the address is shown as all zeros.
Out-Filter	Whether an outbound filter is configured for traffic to the destination network number or node. The value can be one of the following: • No • Yes
Frame-Type	The frame encapsulation type, which can be one of the following: • Ethernet SNAP • Ethernet 802.2 • Ethernet 802.3 • Ethernet II
Port	The port through which the Layer 3 Switch sends traffic to the destination network and node.

USING THE WEB MANAGEMENT INTERFACE

To display the IPX forwarding cache:

1. Log on to the device using a valid user name and password for read-only or read-write access. The System configuration dialog is displayed.

2. Click on the plus sign next to Monitor in the tree view to display the list of monitoring options.
3. Click on the plus sign next to IPX in the tree view to expand the list of IPX option links.
4. Click on the Cache link.

Displaying the IPX Route Table

To display the IPX route table, use one of the following methods.

USING THE CLI

To display the IPX route table, enter the following command at any CLI level:

```
BigIron> show ipx route

Total number of IPX route entries 3

Forwarding

Index   Network   Router          Hops  Ticks Port
1       11110007  0000.0000.0000     0     1     7
2       32D564FA  00a0.24bf.89ca     1     2     5
3       11110005  0000.0000.0000     0     1     5
```

Syntax: show ipx route [<num(hex)>]

The <num(hex)> parameter lets you specify an IPX network number.

This display shows the following information.

Table 22.4: CLI Display of IPX Route Table

This Field...	Displays...
Total number of IPX route entries	The number of entries in the table.
Index	The index number of the table entry.
Network	The IPX network at the route's destination.
Router	The MAC address of the next-hop IPX router.
Hops	The number of hops (routers) separating the Foundry Layer 3 Switch from the network.
Ticks	The number of ticks.
Port	The port through which the Layer 3 Switch sends traffic to the destination network.

USING THE WEB MANAGEMENT INTERFACE

To display the IPX route table:

1. Log on to the device using a valid user name and password for read-only or read-write access. The System configuration dialog is displayed.
2. Click on the plus sign next to Monitor in the tree view to display the list of monitoring options.
3. Click on the plus sign next to IPX in the tree view to expand the list of IPX option links.
4. Click on the Route link.

Displaying the IPX Server Table

To display the IPX server table, use one of the following methods.

USING THE CLI

To display the IPX server table, enter the following command at any CLI level:

```
BigIron> show ipx servers

Total number of IPX server entries 3

Index  Network   Node            Socket    Type      Hops
1      32D564FA  0000.0000.0001   0005     026B        1
         Server-name: FoundryD
2      32D564FA  0000.0000.0001   4006     0278        1
         Server-name: FoundryM
3      32D564FA  0000.0000.0001   0451     0004        1
         Server-name: Foundry-MPR2
```

Syntax: show ipx servers [<name>]

The <name> parameter lets you specify a server name.

This display shows the following information.

Table 22.5: CLI Display of IPX Server Table

This Field...	Displays...
Index	The index number of the table entry.
Network	The network in which the server is located.
Node	The six-byte node number. The node number can be a MAC address or, for some IPX server types, a “1”.
Socket	The two-byte socket number.
Type	The two-byte number for the server type.
Hops	The number of IPX router hops to the server’s network.
Server-name	The IPX server name.

USING THE WEB MANAGEMENT INTERFACE

To display the IPX server table:

1. Log on to the device using a valid user name and password for read-only or read-write access. The System configuration dialog is displayed.
2. Click on the plus sign next to Monitor in the tree view to display the list of monitoring options.
3. Click on the plus sign next to IPX in the tree view to expand the list of IPX option links.
4. Click on the Server link.

Displaying IPX Traffic Statistics

To display IPX traffic statistics, use one of the following methods.

USING THE CLI

To display IPX traffic statistics, enter the following command at any CLI level:

```
BigIron> show ipx traffic

                                                 Dropped              Filtered
Port    Forward    Receive    Transmit    Receive    Transmit   Receive  Transmit
 1/5          46         36           8          2           0         0         0
 1/7           0          0           6          0           0         0         0
Tot           46         36          14          2           0         0         0
```

Syntax: show ipx traffic

This display shows the following information.

Table 22.6: CLI Display of IPX Traffic Statistics

This Field...	Displays...
Port	The port for which the statistics apply. Only the ports that have IPX interfaces configured on them are listed.
Forward	The number of IPX packets received by the Layer 3 Switch from another device and then sent on the port.
Receive	The number of IPX packets received on the port.
Transmit	The number of IPX packets originated on the Layer 3 Switch and sent on the port.
Dropped Receive	The number of packets received on this port by the Layer 3 Switch that the Layer 3 Switch dropped.
Dropped Transmit	The number of packets queued for sending on this port by the Layer 3 Switch but then dropped.
Filtered Receive	The number of packets received by this port that matched an inbound IPX filter configured on the port.
Filtered Transmit	The number of packets queued for sending on this port that matched an outbound IPX filter configured on the port.

USING THE WEB MANAGEMENT INTERFACE

To display summary IPX traffic statistics:

1. Log on to the device using a valid user name and password for read-only or read-write access. The System configuration dialog is displayed.
2. Click on the plus sign next to Monitor in the tree view to display the list of monitoring options.
3. Click on the plus sign next to IPX in the tree view to expand the list of IPX option links.
4. Click on the Traffic link.

This display shows the following information.

Table 22.7: Web Display of IPX Traffic Statistics

This Field...	Displays...
In Packets	The number of IPX packets received on the router.
Out Packets	The number of IPX packets originated on the router and sent on the router.
Forwarding Packets	The number of IPX packets received by the router from another device and then sent on the router.
Rcv Drop Packets	The number of packets received by the router that the router dropped.
Tx Drop Packets	The number of packets queued for sending by the router but then dropped.
Rcv Filter Packets	The number of packets received by the router that matched an inbound IPX filter.
Tx Filter Packets	The number of packets queued for sending that matched an outbound IPX filter.

To display traffic statistics for each port or virtual interface on which an IPX interface is configured:

1. Log on to the device using a valid user name and password for read-only or read-write access. The System configuration dialog is displayed.
2. Click on the plus sign next to Monitor in the tree view to display the list of monitoring options.
3. Click on the plus sign next to IPX in the tree view to expand the list of IPX option links.
4. Click on the Port Counter link.

This display shows the following information.

Table 22.8: Web Display of IPX Port Statistics

This Field...	Displays...
Port	The port or virtual interface on which the IPX interface is configured.
Forward Packets	The number of IPX packets received by the Layer 3 Switch from another device and then sent on the port.
Rcv Packets	The number of IPX packets received on the port.
Tx Packets	The number of IPX packets originated on the Layer 3 Switch and sent on the port.
Rcv Drop Packets	The number of packets received on this port by the Layer 3 Switch that the Layer 3 Switch dropped.
Tx Drop Packets	The number of packets queued for sending on this port by the Layer 3 Switch but then dropped.
Rcv Filter Packets	The number of packets received by this port that matched an inbound IPX filter configured on the port.

Table 22.8: Web Display of IPX Port Statistics (Continued)

This Field...	Displays...
Tx Filter Packets	The number of packets queued for sending on this port that matched an outbound IPX filter configured on the port.

Chapter 23
Configuring AppleTalk

This chapter describes how to configure AppleTalk on Foundry Layer 3 Switches using the CLI and the Web management interface. Foundry Layer 3 Switches support Phase II of AppleTalk routing.

For complete syntax information for the CLI commands shown in this chapter, see the *Foundry Switch and Router Command Line Interface Reference*.

NOTE: In addition to the routing features described in this chapter, the Chassis Layer 3 Switches and the TurboIron/8 support AppleTalk cable VLANs. If you configure multiple cable VLANs, the Layer 3 Switch bridges traffic within a VLAN and routes traffic between VLANs. See the "Configuring AppleTalk Cable VLANs" section in the "Configuring VLANs" chapter of the *Foundry Switch and Router Installation and Basic Configuration Guide*.

Overview of AppleTalk

AppleTalk inter-networks are built upon distinct networks interconnected by routers such as Foundry's NetIron, TurboIron/8, and BigIron. Each network is composed of nodes—workstations, printers, and servers. AppleTalk zones are assigned across AppleTalk networks to further define end-user access to shared resources such as printers and servers.

Address Assignment

AppleTalk node addresses are assigned dynamically. When a Macintosh running AppleTalk starts up, it selects a network address and checks to see if that address is already in use. If the address is already in use by another client, a message will be returned to the requesting station and the process will repeat until an uncommitted address is located.

Network Components

Nodes

The ***node*** is the primary building block of any AppleTalk network. A node is any device on an AppleTalk network such as a workstation, printer, or server running AppleTalk.

Networks

Multiple nodes that share the same logical segment comprise an AppleTalk network. Each node in the network is assigned an AppleTalk address.

An AppleTalk address is comprised of a 16-bit network number and an 8-bit node number. For example, 500.50 refers to node 50 on network 500.

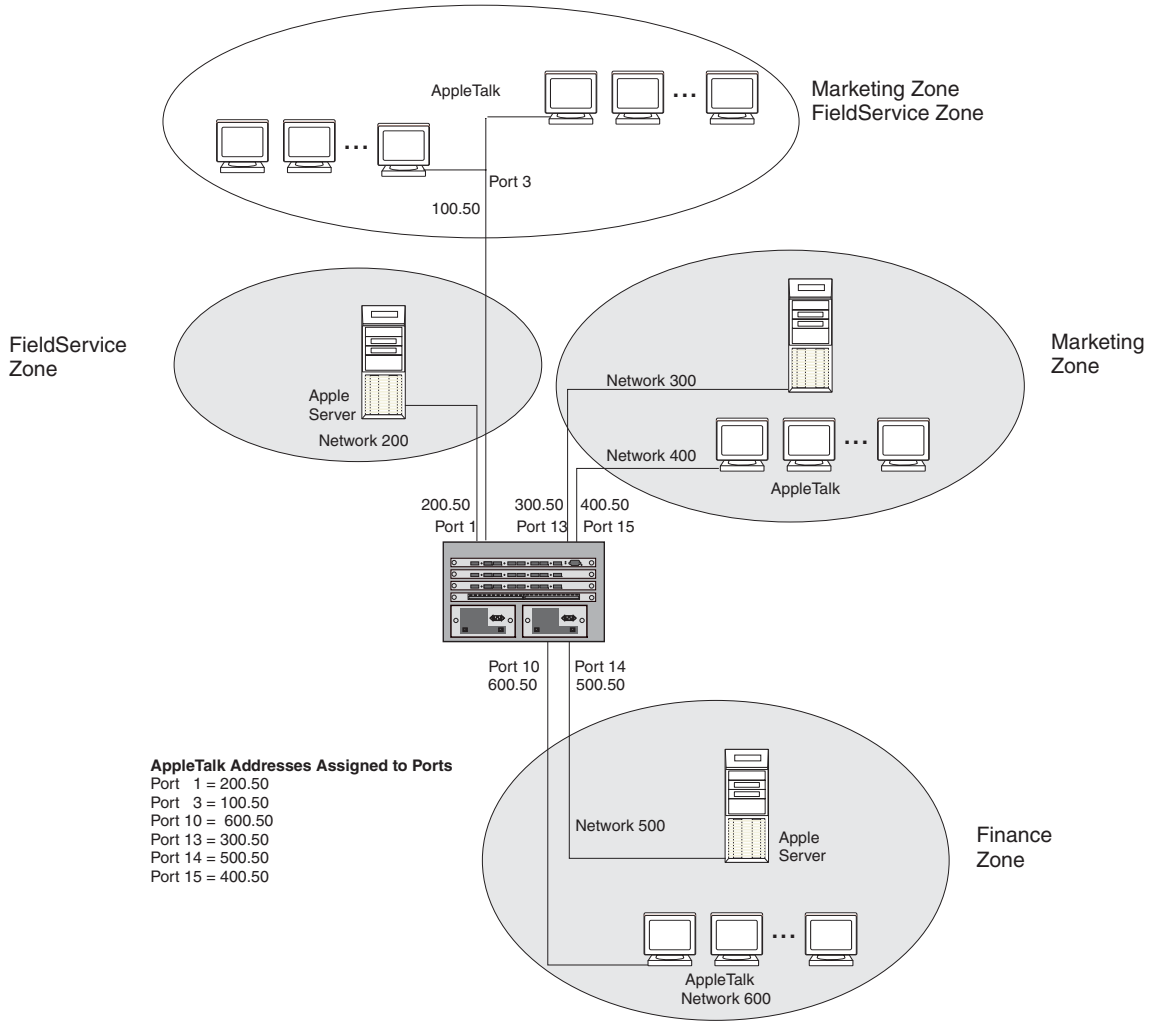
An AppleTalk network address is a single 16-bit network number or a network range (cable range). The network range specifies a range of contiguous network numbers with start and end values.

Zones

AppleTalk zones are logical groupings of AppleTalk nodes defined within and across multiple networks as shown in Figure 23.1. For example, the Finance zone comprises two separate networks, 500 and 600. These network numbers are assigned to a specific interface on a router, and nodes within those networks are automatically assigned numbers in that range.

Defining zones for certain workstations and resources on the network allows you to easily permit or deny access to certain devices or information on the network by providing or hiding information about zones to a node or network. This is further explained in the following sections on filtering.

Figure 23.1 AppleTalk Zones defined within and across AppleTalk networks



Zone Filtering

Zone filtering allows you to define access for a network and its nodes by defining single permit or deny filters, rather than defining an access list for each node independently.

By eliminating the need to enter separate numbers for each device or network segment, zone filters improve overall system administration of an AppleTalk network. For example, if a new device such as a server or laser

printer is added to an existing zone, all users in that zone automatically have access to that device without any additional configuration.

Additionally, this feature helps eliminate unauthorized access to devices within restricted zones. As new devices are added to secured zones, information on those devices is protected automatically.

Network Filtering

You also can filter on a network basis by enabling the Routing Table Maintenance Protocol (RTMP) filtering capability of zone filtering. When this filter is enabled on an interface, the denied network numbers are removed from the RTMP packet before it is transmitted out of the interface.

You can define deny or permit zone and network filters for AppleTalk on an interface basis. You can define up to 32 filters for routers operating with 32MB of memory. For those systems with 8MB of memory installed, you can define up to 16 filters.

Seed and Non-Seed Routers

An AppleTalk router must be configured as either a seed or a non-seed router.

When you configure an AppleTalk router as a seed router, you must define the cable-range, address, and zone names for the router. When you configure a non-seed router, the router will learn its parameters from a seed AppleTalk router on the same segment.

AppleTalk Components Supported on Foundry Layer 3 Switches

The following sections describe the AppleTalk protocol components supported by Foundry Layer 3 Switches.

Session Layer Support

The ***Zone Information Protocol (ZIP)*** maintains the mapping between defined network numbers and zone names within an AppleTalk network. This information is stored on a router in the zone information table.

ZIP also uses information from the RTMP routing table to stay current on the network topology.

Transport Layer Support

Routing Table Maintenance Protocol (RTMP)

RTMP establishes and maintains the AppleTalk routing table. AppleTalk routers use RTMP to exchange routing information at regular intervals to ensure that each router has the latest routing information.

For Foundry Layer 3 Switches, the periodic updates are sent out every 10 seconds by default.

AppleTalk Echo Protocol (AEP)

AppleTalk routers use AEP to check connectivity to other devices on the network.

AppleTalk Transaction Protocol (ATP)

ATP facilitates transaction-based applications. ATP supports a client/server design in which clients request information and servers reply with a response to that request. The protocol assigns a transaction ID to each request/response pair and allows only one instance of that specific transaction.

A sub-set of ATP is implemented to support ZIP on Foundry Layer 3 Switches.

Name Binding Protocol (NBP)

NBP maps AppleTalk names used on a network with addresses. For example, a printer for the marketing group may be named MKTG with an address of 100.5. This association is mapped together by the NBP.

NBP is dynamically initiated when the node is started. NBP also addresses registration, deletion, confirmation, and search of names.

Network Layer Support

Datagram Delivery Protocol (DDP)

DDP provides connectionless service between application sockets on an AppleTalk network and administers AppleTalk addresses.

AppleTalk Address Resolution Protocol (AARP)

AARP translates AppleTalk addresses into 48-bit data link addresses. The 48-bit data link address is required in order to send AppleTalk packets to a specific node. AARP is also used to check for duplicate AppleTalk addresses on the network.

An AARP entry notes the mapping between a node's AppleTalk address and its MAC (hardware) address.

Data Link Support

AppleTalk supports the ***EtherTalk Link Access Protocol (ELAP)***, which defines the Layer 2 encapsulation for AppleTalk packets.

Dynamic AppleTalk Activation and Configuration

AppleTalk is automatically activated when you enable the protocol on systems running software release 4.0 or later. On platforms running an earlier software release, you must reset the system to initially enable AppleTalk; however, all changes after that occur dynamically.

Configuring AppleTalk Routing

To begin using AppleTalk on a Foundry Layer 3 Switch, perform the following tasks:

1. Enable AppleTalk on the router, if it is not already enabled.
2. Configure AppleTalk as either a seed or a non-seed router.

   When you configure a seed router, you define the cable-range, address, and zone names for the router. When you configure a non-seed router, the router will learn its parameters from another AppleTalk router on the same segment.
3. Define zone and additional zone filters, if desired.
4. Configure virtual interfaces to allow routing between AppleTalk VLANs, if desired.
5. Modify global parameters, if desired.

Enable AppleTalk

To enable AppleTalk routing on a router, use one of the following methods.

NOTE: AppleTalk is automatically activated when you enable the protocol on systems running software release 4.0 or later. On platforms running an earlier software release, you must reset the system to initially enable AppleTalk; however, all changes after that occur dynamically.

NOTE: Once AppleTalk is enabled at the global (system) level, no additional configuration is required at this level unless the default parameters assigned need to be modified to address network requirements. See “Modifying AppleTalk Global Parameters” on page 23-17.

USING THE CLI

```
BigIron(config)# router appletalk
BigIron(config)# write memory
BigIron(config)# end
```

Syntax: router appletalk

USING THE WEB MANAGEMENT INTERFACE

1. Log on to the device using a valid user name and password for read-write access. The System configuration dialog is displayed.
2. Select the Enable radio button next to AppleTalk.
3. Click the Apply button to apply the changes to the device's running-config file.
4. 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.

Configuring a Seed AppleTalk Router

When you configure an AppleTalk router as a seed router, you must define the cable range, AppleTalk address, and zone names for the router interfaces.

To configure a seed router, perform the following tasks:

1. Configure the cable range (network numbers) to be supported on that interface.
2. Assign an AppleTalk address to the interface.
3. Assign a zone or zones to the interface.
4. Enable AppleTalk routing on the interface.

NOTE: Before configuring interface parameters for AppleTalk, you must enable AppleTalk at the system level.

USING THE CLI

This section describes defining a cable range, assigning network addresses and zones, and enabling AppleTalk routing on an interface.

Configuring the Cable Range for an Interface

To support network numbers from 10 – 50 on interface 1/3, enter the following commands:

```
BigIron(config)# int e 1/3
BigIron(config-if-1/3)# appletalk cable 10 - 50
```

Syntax: appletalk cable <network-number> | <network-number - network-number>

Configuring a Network Address for an Interface

To assign an AppleTalk address of 10.5 to interface 1/3, enter the following command:

```
BigIron(config-if-1/3)# appletalk address 10.5
```

Syntax: appletalk address <node.network>

Configuring Zones on an Interface

To assign sales, marketing, and finance zones for interface 1/3, enter the following commands:

```
BigIron(config-if-1/3)# appletalk zone sales
BigIron(config-if-1/3)# appletalk zone marketing
BigIron(config-if-1/3)# appletalk zone finance
```

NOTE: Chassis devices and the TurboIron/8 router can support up to 1,536 zones per system and up to 64 per interface. For Stackable devices operating with 32MB of memory, you can define up to 255 zones on the network, and up to 64 on an interface.

For systems with 8MB of memory installed, up to 64 zones can be supported within the network, and up to 8 zones can be defined on an interface basis.

Enabling AppleTalk Routing on an Interface

To enable AppleTalk routing on interface 3, enter the following command:

```
BigIron(config-if-3)# appletalk routing
```

Saving Configuration Changes to the Interface

Once you have configured the cable range, network address, zone(s), and AppleTalk routing for an interface, you can preserve the configuration changes by saving them to flash.

If the system is operating with release 3.0 or earlier software, you also need to reset the system using the reload CLI command, if AppleTalk is not yet enabled. If AppleTalk is active on the router, then no reset of the system is needed.

```
BigIron(config-if-3)# write memory
BigIron(config-if-3)# end
BigIron# reload
```

NOTE: When there is more than one seed router on the network, make sure the AppleTalk configuration of each of those seed routers is consistent with other routers on the same segment.

USING THE WEB MANAGEMENT INTERFACE

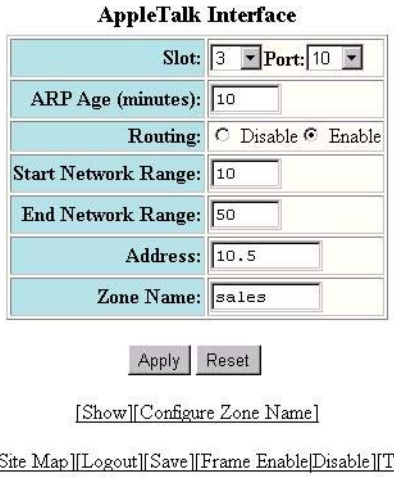
This section describes how to enable AppleTalk on the router as well as how to configure the cable range, network address, and zones for an AppleTalk seed router.

To enable AppleTalk on the router:

1. Log on to the device using a valid user name and password for read-write access. The System configuration dialog is displayed.
2. Select the Enable radio button next to AppleTalk.
3. Click the Apply button to apply the changes to the device's running-config file.
4. 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.

To configure an interface as a seed router:

1. Log on to the device using a valid user name and password for read-write access. The System configuration dialog is displayed.
2. Click on the plus sign next to Configure in the tree view to expand the list of configuration options.
3. Click on the plus sign next to AppleTalk in the tree view to expand the list of AppleTalk option links.
4. Click on the Interface link to display the AppleTalk Interface table.
5. Click on the Modify button next to the interface you want to configure for AppleTalk. The AppleTalk Interface configuration panel is displayed, as shown in the following example.



6. Select the port or slot/port to be configured from the port pulldown menu(s).
7. Modify the ARP age value from the default value of 10 minutes, if desired. Possible values are 1 – 240 minutes.
8. Beginning in software release 06.0.00, the AppleTalk ARP age is a global parameter instead of an interface parameter. When you enter an ARP age value for a port and apply the change to the running-config file or save the change to the startup-config file, the change is saved as the global setting. If you try to set different values for different ports, the interface does not display an error message. Instead, the most recent value you enter before saving the configuration change becomes the global setting.
9. Enable the routing option.
10. Configure the range of supported network addresses by entering the lowest supported number in the Start Network Range field and the highest supported number in the End Network Range field.
11. Enter the AppleTalk address for the port. The address should be a two decimal number, and the first number should be within the network range entered in step 10 above.
12. Enter a zone name for the port.

NOTE: If you do not enter any values other than zero in the network range or address field, and the zone name field is empty, the router will be a non-seed router.

---

13. Click the Apply button to apply the changes to the device's running-config file.
14. 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.

Configuring a Non-seed AppleTalk Router

This section describes how to configure a non-seed router using the CLI or the Web management interface.

To configure a non-seed router, perform the following tasks:

1. Verify that at least one AppleTalk router in the network of the router being configured is operating as a seed router.

NOTE: This requirement ensures that the non-seed router has a seed router on the same segment, from which it can learn configuration details.

---

2. Enable AppleTalk at the global level.
3. Enable AppleTalk routing on the interface(s).

Enabling AppleTalk Routing at the Global (System) Level

To enable AppleTalk on the router, use one of the following methods:

USING THE CLI

```
BigIron(config)# router appletalk
```

USING THE WEB MANAGEMENT INTERFACE

1. Log on to the device using a valid user name and password for read-write access. The System configuration dialog is displayed.
2. Select the Enable radio button next to AppleTalk.
3. Click the Apply button to apply the changes to the device's running-config file.
4. 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.

Enable AppleTalk Routing on an Interface

To enable AppleTalk on interface 1/5, use one of the following methods.

USING THE CLI

```
BigIron(config)# int e 1/5
BigIron(config-if-1/5)# appletalk routing
BigIron(config-if-1/5)# end
BigIron# write memory
BigIron# reload
```

NOTE: When you first enable AppleTalk on a system operating with release 3.0 software or earlier, you must reset (reboot) the system using the **reload** command. All changes after that are dynamic and take effect immediately.

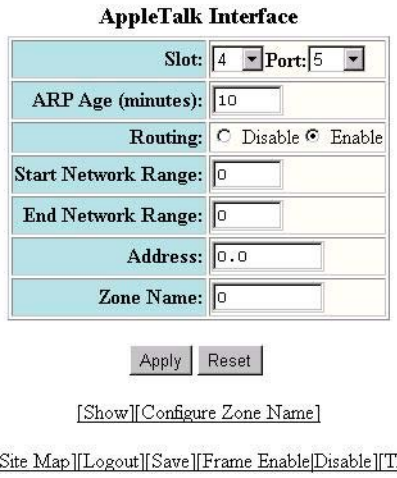
NOTE: By definition, values for the network range, AppleTalk address, and zone name fields are never entered for a non-seed router. If you enter information into these fields, the router is a seed router.

NOTE: Once configured as a non-seed router, the router will send out a query to a seed router on its network to obtain configuration details such as network range, AppleTalk address, and zone name(s) for the router.

USING THE WEB MANAGEMENT INTERFACE

To configure an interface as a non-seed router:

1. Log on to the device using a valid user name and password for read-write access. The System configuration dialog is displayed.
2. Click on the plus sign next to Configure in the tree view to expand the list of configuration options.
3. Click on the plus sign next to AppleTalk in the tree view to expand the list of AppleTalk option links.
4. Click on the Interface link to display the AppleTalk Interface table.
5. Click on the Modify button next to the interface you want to configure for AppleTalk. The AppleTalk Interface configuration panel is displayed, as shown in the following example.



6. Select the port or slot/port to be configured from the port pulldown menu(s).
7. Modify the ARP age value from the default value of 10 minutes, if desired. Possible values are 1 – 240 minutes.
8. Beginning in software release 06.0.00, the AppleTalk ARP age is a global parameter instead of an interface parameter. When you enter an ARP age value for a port and apply the change to the running-config file or save the change to the startup-config file, the change is saved as the global setting. If you try to set different values for different ports, the interface does not display an error message. Instead, the most recent value you enter before saving the configuration change becomes the global setting.
9. Enable the routing option.
10. Click the Apply button to apply the changes to the device's running-config file.
11. 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.

Modifying AppleTalk Interface Configurations

Once AppleTalk is active on a router, all configuration changes are dynamic and require no reset. However, once you configure an interface for AppleTalk, you must disable AppleTalk routing before you can make any changes to the cable range, network address, or zones values. Once you make changes, you then must re-enable AppleTalk routing for the new changes to take effect.

EXAMPLE:

Suppose you want to expand the network numbers supported on interface 1/3 from the range 10 – 50 to the range 10 – 100. Additionally, you want to add engineering and human resource zones to the interface. To do so, use one of the following methods.

USING THE CLI

```
BigIron(config)# int e 1/3
BigIron(config-if-1/3)# no appletalk routing
BigIron(config-if-1/3)# appletalk cable 10-100
BigIron(config-if-1/3)# appletalk zone engineering
BigIron(config-if-1/3)# appletalk zone humanresource
BigIron(config-if-1/3)# appletalk routing
BigIron(config-if-1/3)# end
BigIron# write memory
```


USING THE WEB MANAGEMENT INTERFACE

1. Log on to the device using a valid user name and password for read-write access. The System configuration dialog is displayed.
2. Click on the plus sign next to Configure in the tree view to expand the list of configuration options.
3. Click on the plus sign next to AppleTalk in the tree view to expand the list of AppleTalk option links.
4. Click on the Interface link to display the AppleTalk Interface table.
5. Click on the Modify button next to the interface you want to reconfigure for AppleTalk. The AppleTalk Interface configuration panel is displayed.
6. Modify parameters as needed.
7. To modify other interfaces, select the port (and slot number if applicable) from the Port and Slot fields, then modify the values.
8. Click the Apply button to apply the changes to the device's running-config file.
9. 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.

Filtering AppleTalk Zones and Networks

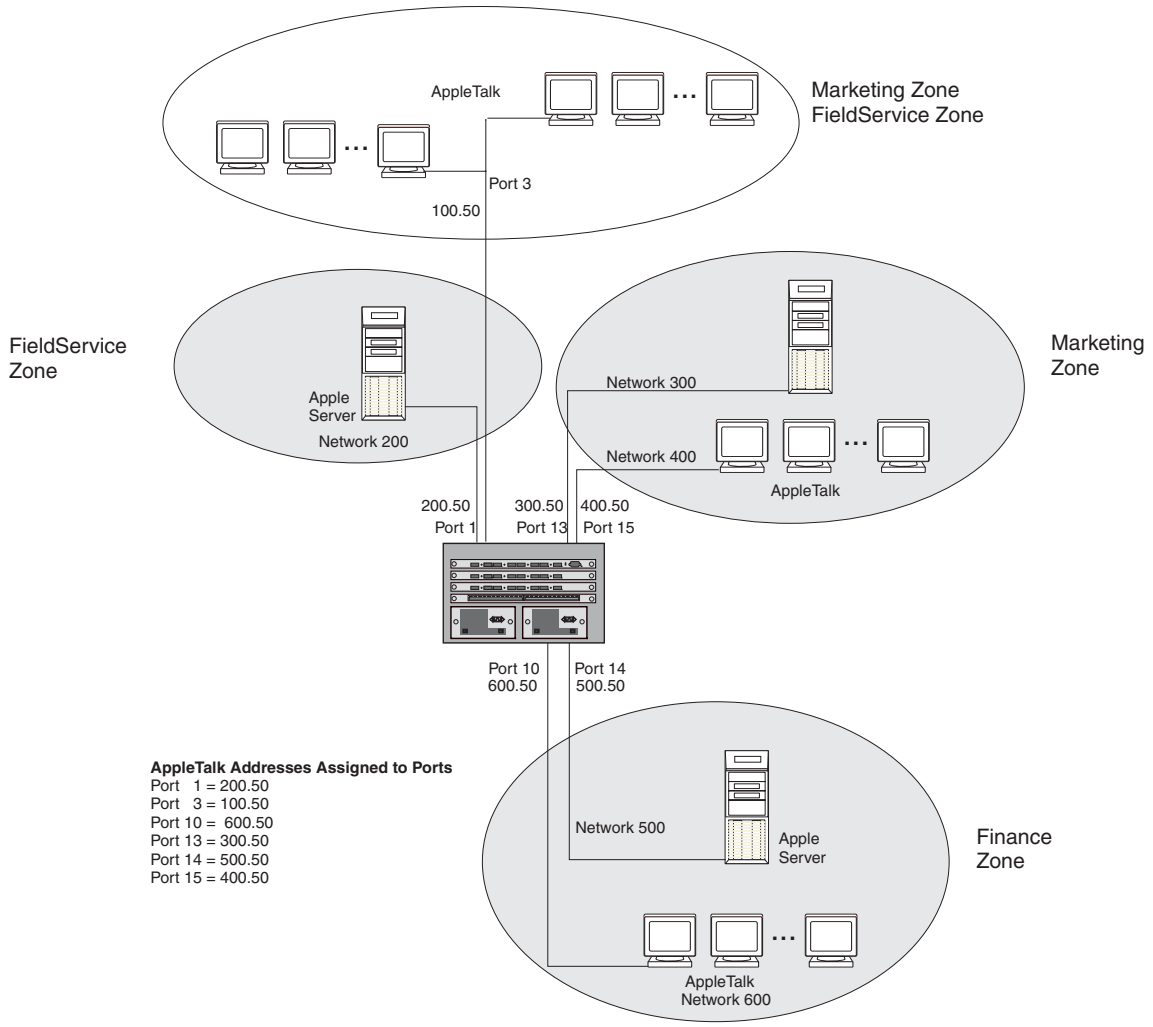
Defining Zone Filters

Zone filtering allows you to define access for a network and its nodes by entering single permit or deny CLI commands, instead of defining an access list for each node independently.

By eliminating the need to enter separate numbers for each device or network segment, zone filters improve overall system administration of an AppleTalk network. For example, if a new device such as a server or laser printer is added to an existing zone, all users in that zone automatically have access to that device without any additional configuration.

Additionally, zone filters help eliminate unauthorized access to devices within restricted zones. As new devices are added to secured zones, information on those devices is protected automatically.

Figure 23.2 AppleTalk zones in a network



EXAMPLE:

Suppose you want to deny access to the Finance server to users within the Marketing and Field Service zones on the network, as shown in Figure 23.2. To define a zone filter for this, use one of the following methods.

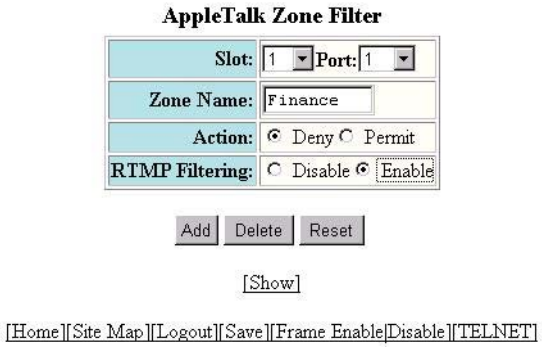
USING THE CLI

```
BigIron(config)# interface e 1/1
BigIron(config-if-1/1)# appletalk deny zone finance
BigIron(config-if-1/1)# int e 1/3
BigIron(config-if-1/3)# appletalk deny zone finance
BigIron(config-if-1/3)# int e 1/13
BigIron(config-if-1/13)# appletalk deny zone finance
BigIron(config-if-1/13)# int e 1/15
BigIron(config-if-1/15)# appletalk deny zone finance
```

USING THE WEB MANAGEMENT INTERFACE

1. Log on to the device using a valid user name and password for read-write access. The System configuration dialog is displayed.
2. Click on the plus sign next to Configure in the tree view to expand the list of configuration options.
3. Click on the plus sign next to AppleTalk in the tree view to expand the list of AppleTalk option links.

4. Click on the Zone Filter link.
   - If the device does not have any AppleTalk zone filters, the AppleTalk Zone Filter configuration panel is displayed, as shown in the following example.
   - If an AppleTalk zone filter is already configured and you are adding a new one, click on the Configure AppleTalk Zone Filter link to display the AppleTalk Zone Filter configuration panel, as shown in the following example.
   - If you are modifying an existing AppleTalk zone filter, click on the Modify button to the right of the row describing the filter to display the AppleTalk Zone Filter configuration panel, as shown in the following example.



5. Select the interface for which a zone filter is to be defined, from the port or slot/port pull down menu(s). In this example, you are defining a zone filter for interfaces 1, 3, 13, and 15, all of which have membership in either or both of the Marketing and Field Service zones.
6. Enter the name of the zone to which you are permitting or denying access. In this case, enter Finance.
7. Select either Deny or Permit. In this example, select Deny for interfaces 1, 3, 13, and 15.
8. Enable RTMP filtering to filter on a network basis. When RTMP filtering is enabled on an interface, the denied network numbers are removed from the RTMP packet before it is transmitted out of the interface.
9. Click the Apply button to apply the changes to the device's running-config file.
10. 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.

Define Additional Zone Filters

When defining AppleTalk zone filters, you must define both deny and permit relationships for an interface. For instance, in the previous example, a deny filter prevents users within Marketing and Field Service zones from accessing the Finance zone.

Because all additional zones not specifically addressed by a deny filter are permitted by default, you do not need to configure any specific permit definitions, and the requirement of defining both deny and permit relationships is satisfied.

However, the additional zone filter is useful in denying access to those zones not specifically addressed in permit zone filters. Consider the following example.

EXAMPLE:

Suppose Sales, Human Resources (HR), Engineering, and Training zones will be added to the network in the next month. You know in advance that the only other zone that will be allowed access to the Finance zone is the HR zone.

You can configure permit zone filters for ports 10 and 14 that allow the HR zone to have access to the finance zone and deny access to all others with a deny additional zone filter. This approach addresses the current network and all future zone additions with no additional configuration.

USING THE CLI

To define the permit filter for HR on ports 1/10 and 1/14, enter the following commands:

```
BigIron(config)# interface e 1/10
BigIron(config-if-1/10)# no appletalk routing
BigIron(config-if-1/10)# appletalk permit zone HR
BigIron(config-if-1/10)# deny additional-zones
BigIron(config-if-1/10)# appletalk routing
BigIron(config-if-1/10)# int e 1/14
BigIron(config-if-1/14)# no appletalk routing
BigIron(config-if-1/14)# appletalk permit zone HR
BigIron(config-if-1/14)# appletalk routing
BigIron(config-if-1/14)# write memory
```

NOTE: You must disable AppleTalk routing on any interface already operating with AppleTalk before making any modifications to the configuration, and then re-enable routing to activate the change.

USING THE WEB MANAGEMENT INTERFACE

To define the permit and deny filters discussed above:

1. Log on to the device using a valid user name and password for read-write access. The System configuration dialog is displayed.
2. Click on the plus sign next to Configure in the tree view to expand the list of configuration options.
3. Click on the plus sign next to AppleTalk in the tree view to expand the list of AppleTalk option links.
4. Click on the Zone Filter link.
   - If the device does not have any AppleTalk zone filters, the AppleTalk Zone Filter configuration panel is displayed.
   - If an AppleTalk zone filter is already configured and you are adding a new one, click on the Configure AppleTalk Zone Filter link to display the AppleTalk Zone Filter configuration panel.
   - If you are modifying an existing AppleTalk zone filter, click on the Modify button to the right of the row describing the filter to display the AppleTalk Zone Filter configuration panel.
5. Select the interface for which the zone filter is to be defined from the port or slot/port pull down menu(s). In this example, you are defining a permit zone filter for HR for interfaces 10 and 14, which have membership in the Finance zone.
6. Enter the zone name to which access is to be permitted or denied. In this case, the zone name is HR.
7. Select either Deny or Permit. In this example, select Permit for interfaces 10 and 14.
8. Enable RTMP filtering to also filter on a network basis.

   NOTE: When this filter is enabled on an interface, the denied network numbers are removed from the RTMP packet before it is transmitted out of the interface. In this example, RTMP filtering is not desired, so this option default is left as disabled.

9. Click the Apply button to apply the changes to the device's running-config file.
10. Click on the Additional Zone Filter link in the tree view.
11. Select the interface for which the zone filter is to be defined, from the port or slot/port pull down menu(s). In this example, define a deny zone filter for interfaces 10 and 14 to deny all other zones not specified in the permit zone filter (steps 1 – 6 above).
12. Select either Deny or Permit. For this example, select Deny for interfaces 10 and 14.
13. Disable RTMP filtering.
14. Click the Apply button to apply the changes to the device's running-config file.

15. 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.

Network Filtering

EXAMPLE:

To deny access to the Finance server to users within the Marketing and Field Service zones on the network and to prevent information about the zone and the network numbers from being forwarded out of interface 1 (Figure 23.2), use one of the following methods.

USING THE CLI

```
BigIron(config-if-1/1)# appletalk deny zone finance rtmp-filtering
```

USING THE WEB MANAGEMENT INTERFACE

To enable RTMP filtering on an interface, define the filter as usual, then enable the RTMP filtering option on the AppleTalk Zone Filter panel.

Routing Between AppleTalk VLANs Using Virtual Interfaces

In addition to supporting AppleTalk VLANs, Foundry Layer 3 Switches support routing between AppleTalk VLANs using virtual interfaces. The virtual interfaces provide VLANs access to the router functions of Foundry Layer 3 Switches. Using these virtual interfaces eliminates the need to assign a physical port for routing between local VLANs.

AppleTalk routing between virtual and physical interfaces is also supported.

EXAMPLE:

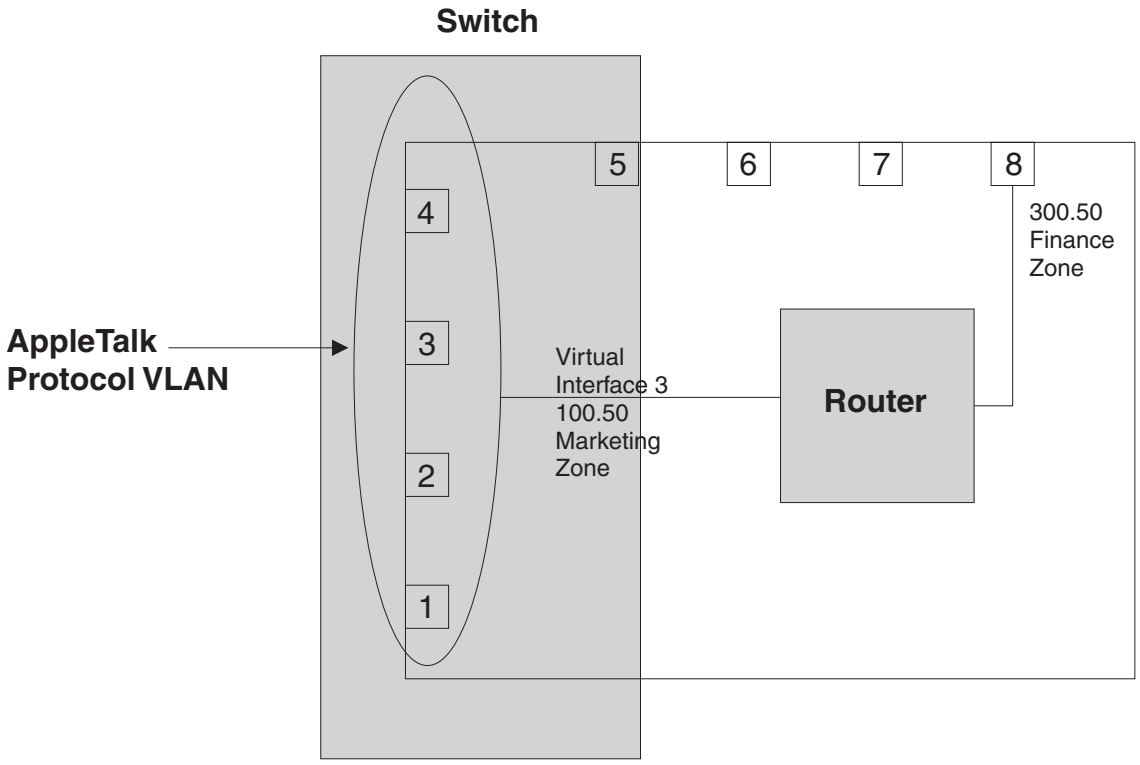
In Figure , AppleTalk traffic is terminating on ports 1 through 4. Suppose you want to group all of these interfaces into an AppleTalk protocol VLAN and route traffic to VLANs on other routers.

To do so, perform the following steps:

1. Create an AppleTalk protocol VLAN with port membership of ports 1, 2, 3, and 4.
2. Assign a virtual interface to the AppleTalk VLAN to allow it to route traffic to AppleTalk VLANs on remote routers.
3. Configure a physical interface on the router that provides access to remote networks to support routing between local and remote AppleTalk VLANs.

NOTE: By supporting assignment of VLANs on interfaces, the router is functioning as a virtual switch.

Figure 23.3 Virtual interface provides a routing interface to an AppleTalk VLAN



USING THE CLI

To configure the AppleTalk VLAN as seen in Figure , enter the following commands:

```
BigIron(config)# router appletalk
BigIron(config)# vlan 1
BigIron(config-vlan-1)# atalk-proto
BigIron(config-vlan-atalk-proto)# static e1 to 4
BigIron(config-vlan-atalk-proto)# router-interface ve 3
```

To configure the physical interface (e 1/8) to which all outgoing traffic is forwarded, enter the following commands:

```
BigIron(config-vlan-atalk-proto)# int e 1/8
BigIron(config-if-1/8)# appletalk cable-range 300 - 300
BigIron(config-if-1/8)# appletalk address 300.50
BigIron(config-if-1/8)# appletalk zone-name Finance
BigIron(config-if-1/8)# appletalk routing
```

To configure the defined AppleTalk VLAN virtual interface ve3, enter the following commands:

```
BigIron(config-if-1/8)# int ve 3
BigIron(config-vif-3)# appletalk cable-range 100 - 100
BigIron(config-vif-3)# appletalk address 100.50
BigIron(config-vif-3)# appletalk zone-name Marketing
BigIron(config-vif-3)# appletalk routing
```

Routing Between Protocol VLANs Within Port-Based VLANs

In Figure 23.4, AppleTalk traffic is terminating on ports 1 through 4 on two separate networks, 100 and 200. Suppose you want to assign these networks to two separate VLANs but would also like to route traffic between the two VLANs and externally to the router.

To create the configuration shown in Figure 23.4, perform the following tasks.

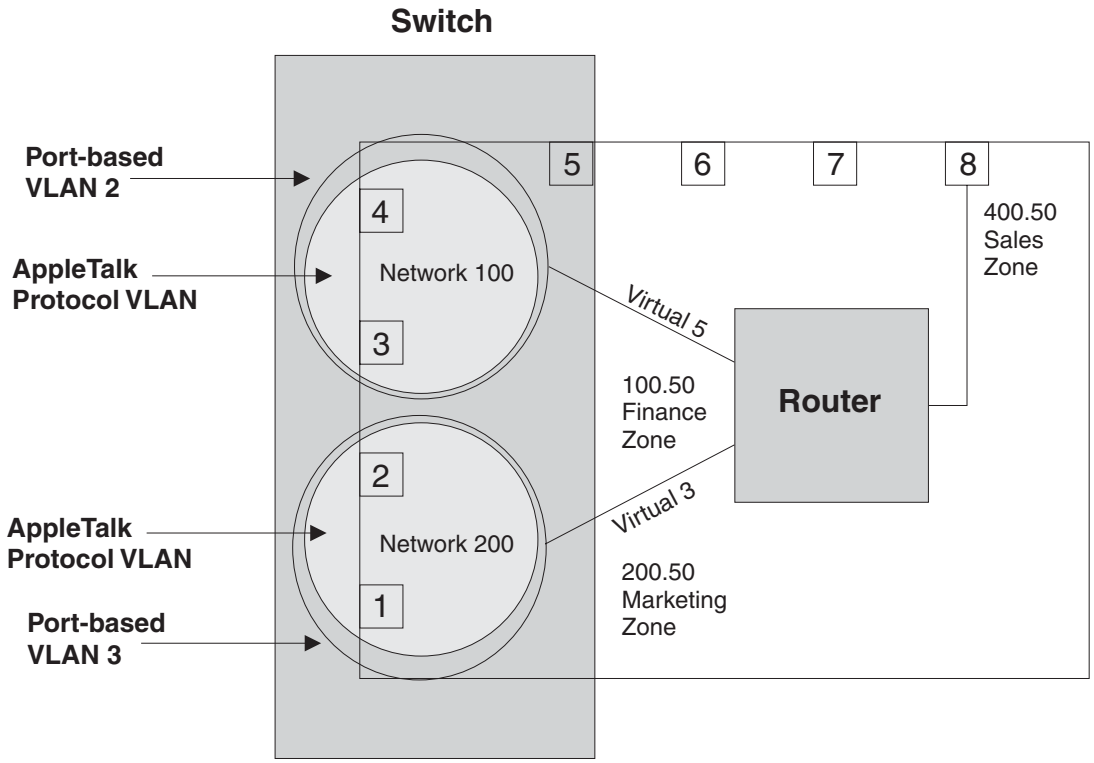
1. Create port-based VLANs 2 and 3.

NOTE: Protocol VLANs must always be within the boundaries of a port-based domain. Whenever port and protocol VLANs operate on a system together, you must create the port-based VLAN before you create the protocol VLAN. The protocol-based VLAN overlays the port-based VLAN.

2. Create AppleTalk protocol VLANs 2 and 3.
3. Configure router interfaces virtual 3 (v3) and virtual 5 (v5).
4. Configure physical interface port 8.

NOTE: Each of the above tasks is described in the following sections.

Figure 23.4 Routing between AppleTalk VLANs



USING THE CLI

```
BigIron(config)# vlan 2 by port
BigIron(config-vlan-2)# untag e3 to 4
BigIron(config-vlan-2)# atalk-proto
BigIron(config-vlan-atalk-proto)# static e3 to 4
BigIron(config-vlan-atalk-proto)# router-interface ve 5
BigIron(config-vlan-atalk-proto)# end
BigIron(config-vlan-2)# vlan 3 by port
BigIron(config-vlan-3)# untag e1 to 2
BigIron(config-vlan-3)# atalk-proto
BigIron(config-vlan-atalk-proto)# router-interface ve 3
```


To configure the physical interface (e 1/8) to which all outgoing traffic is forwarded, enter the following commands:

```
BigIron(config-vlan-atalk-proto)# int e 1/8
BigIron(config-if-1/8)# appletalk cable-range 400 - 400
BigIron(config-if-1/8)# appletalk address 400.50
BigIron(config-if-1/8)# appletalk zone-name sales
BigIron(config-if-1/8)# appletalk routing
```

To configure the defined AppleTalk VLAN virtual interfaces ve3 and ve5, enter the following commands:

```
BigIron(config-if-1/8)# int ve 5
BigIron(config-vif-5)# appletalk cable-range 100 - 100
BigIron(config-vif-5)# appletalk address 100.50
BigIron(config-vif-5)# appletalk zone-name finance
BigIron(config-vif-5)# appletalk routing
BigIron(config-vif-5)# int ve 3
BigIron(config-vif-3)# appletalk cable-range 200 - 200
BigIron(config-vif-3)# appletalk address 200.50
BigIron(config-vif-3)# appletalk zone-name marketing
BigIron(config-vif-3)# appletalk routing
BigIron(config-vif-3)# end
BigIron# write memory
```

Modifying AppleTalk Global Parameters

You can modify the following AppleTalk parameters at the global level:

- AppleTalk ARP age
- AppleTalk ARP retransmission count
- AppleTalk ARP retransmission interval
- AppleTalk glean packets
- AppleTalk QoS socket (assigns a higher priority)
- AppleTalk RTMP update interval
- AppleTalk ZIP query interval

The following sections describe these parameters and show how to change them.

AppleTalk ARP Age

To change the AppleTalk ARP age in software release 06.0.00, use one of the following methods.

USING THE CLI

To change the AppleTalk ARP age, enter the following command at any level of the CLI:

```
BigIron(config)# appletalk arp-age 30
BigIron(config)# write memory
```

Syntax: [no] appletalk arp-age <num>

The <num> parameter specifies the number of minutes for the ARP age and can be from 1 – 240. The default is 10.

USING THE WEB MANAGEMENT INTERFACE

You can change the AppleTalk ARP age using the Web management interface, but the interface still allows you to enter the ARP age value only on an individual port basis. However, when you enter an ARP age value for a port and apply the change to the running-config file or save the change to the startup-config file, the change is saved as the global setting. If you try to set different values for different ports, the interface does not display an error message. Instead, the most recent value you enter before saving the configuration change becomes the global setting.

1. Log on to the device using a valid user name and password for read-write access.
2. If you have not already enabled AppleTalk, enable it by clicking on the Enable radio button next to AppleTalk on the System configuration dialog, then clicking Apply to apply the change.
3. Click on the plus sign next to Configure in the tree view to expand the list of configuration options.
4. Click on the plus sign next to AppleTalk in the tree view to expand the list of AppleTalk option links.
5. Click on the Interface link to display the AppleTalk Interface table.
6. Click on the Modify button to the right of any port listed in the table to display the AppleTalk Interface configuration panel. Regardless of the port you choose, the setting will take effect globally.
7. Edit the value in the ARP Age field to the new ARP age. You can enter a value from 1 – 240 minutes. The default is 10 minutes.
8. Click the Apply button to apply the change to the device's running-config file.
9. Select the Save link at the bottom of the dialog, then select Yes when prompted to save the configuration change to the startup-config file on the device's flash memory.

AppleTalk ARP Retransmit Count

You can modify the maximum number of times that a packet will be sent out for ARP cache informational updates. The packet is sent out to the maximum amount defined, until the information is received.

If no response is received before the count number expires, the router does not send any additional packets. Possible values are from 1 – 10. The default is 2.

EXAMPLE:

To modify the number of times packet requests are sent out for ARP updates from the default (2) to 8, use one of the following methods.

USING THE CLI

```
BigIron(config)# appletalk arp retransmit-count 8
```

Syntax: appletalk arp retransmit-count <1-10>

USING THE WEB MANAGEMENT INTERFACE

1. Log on to the device using a valid user name and password for read-write access.
2. Click on the plus sign next to Configure in the tree view to expand the list of configuration options.
3. Click on the plus sign next to AppleTalk in the tree view to expand the list of AppleTalk option links.
4. Click on the General link to display the AppleTalk configuration panel.
5. Enter a new ARP retransmit count from 1 – 10 in the ARP Retransmit Count field. For this example, enter 8.
6. Click the Apply button to apply the change to the device's running-config file.
7. Select the Save link at the bottom of the dialog, then select Yes when prompted to save the configuration change to the startup-config file on the device's flash memory.

AppleTalk ARP Retransmit Interval

You can modify the interval between the transmission of ARP packets. Possible values are from 1 – 120 seconds. The default is 1 second.

EXAMPLE:

To modify the ARP retransmission interval from the default value (1) to 15 seconds, use one of the following methods.

USING THE CLI

```
BigIron(config)# appletalk arp retransmit-interval 15
```


Syntax: appletalk arp retransmit-interval <1-120>

USING THE WEB MANAGEMENT INTERFACE

1. Log on to the device using a valid user name and password for read-write access.
2. Click on the plus sign next to Configure in the tree view to expand the list of configuration options.
3. Click on the plus sign next to AppleTalk in the tree view to expand the list of AppleTalk option links.
4. Click on the General link to display the AppleTalk configuration panel.
5. Enter a new AppleTalk ARP Retransmit Interval from 1 – 120 in the ARP Retransmit Interval field. For this example, enter 15.
6. Click the Apply button to apply the change to the device's running-config file.
7. Select the Save link at the bottom of the dialog, then select Yes when prompted to save the configuration change to the startup-config file on the device's flash memory

NOTE: You also can access the dialog for saving configuration changes by clicking on Command in the tree view, then clicking on Save to Flash.

AppleTalk Glean Packets

When you enable the glean packet parameter on an AppleTalk router, the router tries to learn the MAC address from the packet instead of sending out an ARP request. The glean packets parameter is disabled by default.

EXAMPLE:

To enable glean packets on an AppleTalk router, use one of the following methods.

USING THE CLI

```
BigIron(config)# appletalk glean-packets
```

Syntax: appletalk glean-packets

USING THE WEB MANAGEMENT INTERFACE

1. Log on to the device using a valid user name and password for read-write access.
2. Click on the plus sign next to Configure in the tree view to expand the list of configuration options.
3. Click on the plus sign next to AppleTalk in the tree view to expand the list of AppleTalk option links.
4. Click on the General link to display the AppleTalk configuration panel.
5. Select Enable next to Glean Packet.
6. Click the Apply button to apply the change to the device's running-config file.
7. Select the Save link at the bottom of the dialog, then select Yes when prompted to save the configuration change to the startup-config file on the device's flash memory.

AppleTalk QoS Socket

You can use the QoS socket parameter to assign a higher priority to specific AppleTalk sockets. Possible values are 0 (normal priority) to 7 (highest priority) for Chassis devices (including the TurboIron/8). For Stackable devices, the options are normal or high. The default value for all sockets is normal (Stackable devices) or 0 (Chassis devices).

For more information and procedures, see "Assigning AppleTalk Sockets to Priority Queues" on page 2-27.

AppleTalk RTMP Update Interval

You can change the RTMP update interval to modify how often the router sends RTMP updates on AppleTalk interfaces. Possible values are from 1 – 3600 seconds. The default is 10 seconds.

EXAMPLE:

To change the value to 50 seconds from a default value of 10 seconds, use one of the following methods.

USING THE CLI

```
BigIron(config)# appletalk rtmp-update-interval 50
```

Syntax: appletalk rtmp-update-interval <1-3600>

USING THE WEB MANAGEMENT INTERFACE

1. Log on to the device using a valid user name and password for read-write access.
2. Click on the plus sign next to Configure in the tree view to expand the list of configuration options.
3. Click on the plus sign next to AppleTalk in the tree view to expand the list of AppleTalk option links.
4. Click on the General link to display the AppleTalk configuration panel.
5. Enter a new RTMP update interval from 1 – 3600 in the RTMP Update Interval field. For this example, enter 50.
6. Click the Apply button to apply the change to the device's running-config file.
7. Select the Save link at the bottom of the dialog, then select Yes when prompted to save the configuration change to the startup-config file on the device's flash memory.

AppleTalk ZIP Query Interval

You can change the ZIP query interval to modify how often the router retransmits ZIP query messages. Possible values are from 1 – 1000 seconds. The default is 10 seconds.

EXAMPLE:

To change the ZIP query interval to 30 seconds from the default value (10 seconds), use one of the following methods.

USING THE CLI

```
BigIron(config)# appletalk zip-query-interval 30
```

Syntax: appletalk zip-query <1-1,000>

USING THE WEB MANAGEMENT INTERFACE

1. Log on to the device using a valid user name and password for read-write access.
2. Click on the plus sign next to Configure in the tree view to expand the list of configuration options.
3. Click on the plus sign next to AppleTalk in the tree view to expand the list of AppleTalk option links.
4. Click on the General link to display the AppleTalk configuration panel.
5. Enter a new ZIP query interval from 1 – 1000 in the ZIP Query Interval field. For this example, enter 30.
6. Click the Apply button to apply the change to the device's running-config file.
7. Select the Save link at the bottom of the dialog, then select Yes when prompted to save the configuration change to the startup-config file on the device's flash memory.

Displaying AppleTalk Information

You can use the CLI or the Web management interface to display configuration information and statistics for AppleTalk.

USING THE CLI

When using the CLI, you can access information about AppleTalk by entering the following **show** commands.

NOTE: For more details on these commands, see the *Foundry Switch and Router Command Line Interface Reference*.

- **show appletalk arp cache**: Displays the ARP table for the AppleTalk routing protocol.
- **show appletalk forward cache**: Displays the forwarding table for the AppleTalk routing protocol.
- **show appletalk routing table**: Displays the global configuration parameters for the AppleTalk routing protocol.
- show appletalk zone table: Displays the network numbers and zones learned on the network.
- **show appletalk interface**: Displays the AppleTalk configuration for an individual interface or all interfaces.
- **show appletalk interface zone**: Displays the zones defined on all AppleTalk interfaces.
- **show appletalk route**: Displays the AppleTalk routing table.
- **show appletalk traffic**: Displays statistical information for RTMP, ZIP, AEP, DDP and AARP packets.

USING THE WEB MANAGEMENT INTERFACE

1. Log on to the device using a valid user name and password for read-only or read-write access. The System configuration dialog is displayed.
2. Click on the plus sign next to Monitor in the tree view to display the monitoring options.
3. Click on the plus sign next to AppleTalk in the tree view to expand the list of AppleTalk option links.
4. Select one of the following links:
   - The ARP Cache link
   - The Forward Cache link
   - The Interface link
   - The Interface Zone link
   - The Routing Table link
   - The Traffic link
   - The Zone Table link

Clearing AppleTalk Information

USING THE CLI

When using the CLI, you can clear AppleTalk data by entering the following CLI commands:

- **clear appletalk arp cache**: Erases all data in the AppleTalk ARP table, as displayed by the **show appletalk arp** command.
- **clear appletalk forward cache**: Erases all learned data from non-local networks that is currently resident in the AppleTalk cache (forwarding table), as displayed by the **show appletalk cache** command.
- clear appletalk route: Erases all learned routes and zones (non-local routes and zones) currently resident in the AppleTalk routing table, as displayed by the show appletalk route command.

- **clear appletalk statistics**: Erases all RTMP, ZIP, AEP, DDP, and AARP statistics for the router. You can display a summary of the statistics that will be erased by entering the **show appletalk traffic** command.

NOTE: For more details on these commands, see the *Foundry Switch and Router Command Line Interface Reference*.

USING THE WEB MANAGEMENT INTERFACE

1. Log on to the device using a valid user name and password for read-write access. The System configuration dialog is displayed.
2. Click on the plus sign next to Command in the tree view to expand the list of command options.
3. Click on the Clear link to display the Clear panel.
4. Select one of the following:
   - AppleTalk ARP Cache
   - AppleTalk Forward Cache
   - AppleTalk Route
   - AppleTalk Statistics
5. Click the Apply button to implement the change.

Chapter 24
Voice Over IP

Starting in Enterprise software release 08.0.00, you can configure the Foundry 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 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 system 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.

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:

```
BigIron(config)# interface e 1/2
```



```
BigIron(config-if-1/2)# voice-vlan 1001
```

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

```
BigIron(config)# interface e 1/1-1/8
BigIron(config-mif-1/1-1/8)# 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, use the **show voice-vlan** <port-num> command. The following example shows the command output results.

```
BigIron(config)# show voice-vlan ethernet 1/2
Voice vlan ID for port 1/2: 1001
```

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

```
BigIron(config)# show voice-vlan ethernet 1/2
Voice vlan is not configured for port 1/2.
```

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

```
BigIron(config)# show voice-vlan
Port ID          Voice-vlan
1/2              1001
1/8              150
1/5              200
```

Syntax: show voice-vlan [<port-num>]

Appendix A
Remote Network Monitoring

This appendix describes the remote monitoring features available on Foundry products:

- Remote Monitoring (RMON) statistics – All Foundry products support RMON statistics on the individual port level. See “RMON Support” on page A-3.
- NetFlow – NetFlow collects statistical data for traffic flows on a Foundry device and exports the flow data to accounting and billing applications. See “NetFlow” on page A-8.
- sFlow – sFlow collects interface statistics and traffic samples from individual interfaces on a Foundry device and exports the information to a monitoring server. See “sFlow” on page A-30.

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.

USING THE CLI

To view the software and hardware details for the system, enter the **show version** command:

```
BigIron# show version
```

Syntax: show version

USING THE WEB MANAGEMENT INTERFACE

1. Log on to the device using a valid user name and password for read-only or read-write access. The System configuration dialog is displayed.
2. Click on the plus sign next to Monitor in the tree view to expand the list of monitoring options.
3. Click on the Device link to display the Device Information panel.

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.

USING THE CLI

To determine the available show commands for the system or a specific level of the CLI, enter the following command:

```
BigIron# show ?
```

Syntax: show <option>

You also can enter “show” at the command prompt, then press the TAB key.

NOTE: For a complete summary of all available **show...** CLI commands and their displays, see the *Foundry Switch and Router Command Line Interface Reference*.

USING THE WEB MANAGEMENT INTERFACE

1. Log on to the device using a valid user name and password for read-only or read-write access. The System configuration dialog is displayed.
2. Click on the plus sign next to Monitor in the tree view to expand the list of monitoring options.
3. If needed, click on the plus sign next to a subcategory to display the monitoring links for that category.
4. Click on the link for the information you want to view.

Viewing Port Statistics

Port statistics are polled by default every 10 seconds.

USING THE CLI

You can view statistics for ports by entering the following **show** commands:

- show interfaces
- show configuration

USING THE WEB MANAGEMENT INTERFACE

To view the port statistics for all ports on a Layer 2 Switch or Layer 3 Switch:

1. Log on to the device using a valid user name and password for read-only or read-write access. The System configuration dialog is displayed.
2. Click on the plus sign next to Monitor in the tree view to expand the list of monitoring options.
3. Click on the plus sign next to Port to expand the list of port monitoring options.
4. Click on the plus sign next to Statistics to expand its options.
5. Select the link to the port type you want (for example, Ethernet) to display the Port Statistics table.

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 modify this polling rate (when using the Web management interface), select the Preferences link from the main menu, and modify the STP field. You can disable polling by setting the field to zero.

USING THE CLI

To view spanning tree statistics, enter the **show span** command. To view STP statistics for a VLAN, enter the **span vlan** command.

USING THE WEB MANAGEMENT INTERFACE

1. Log on to the device using a valid user name and password for read-only or read-write access. The System configuration dialog is displayed.
2. Click on the plus sign next to Monitor in the tree view to expand the list of monitoring options.

3. Select the STP link.

Clearing Statistics

You can clear statistics for many parameters with the clear option.

USING THE CLI

To determine the available **clear** commands for the system, enter the following command:

```
BigIron# clear ?
```

Syntax: clear <option>

You also can enter “clear” at the command prompt, then press the TAB key.

For a complete summary of all available **clear...** CLI commands and their displays, see the *Foundry Switch and Router Command Line Interface Reference*.

NOTE: Clear commands are found at the Privileged EXEC level.

USING THE WEB MANAGEMENT INTERFACE

You can clear statistics by doing the following:

1. Log on to the device using a valid user name and password for read-write access. The System configuration dialog is displayed.
2. Click on the plus sign next to Command in the tree view to expand the list of command options.
3. Click on the Clear link to display the Clear panel.
4. Select all items to be cleared.
5. Click Apply.

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.

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.

USING THE CLI

You can view a textual summary of the statistics for all ports by entering the following CLI command:

```
BigIron(config)# 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 [<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 the product is a Stackable 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 A.1: 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.

Table A.1: Export Configuration and Statistics (Continued)

This Line...	Displays...
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.
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.

Table A.1: Export Configuration and Statistics (Continued)

This Line...	Displays...
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.
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.

USING THE WEB MANAGEMENT INTERFACE

To view the RMON statistics for the system:

1. Log on to the device using a valid user name and password for read-only or read-write access. The System configuration dialog is displayed.
2. Click on the plus sign next to Monitor in the tree view to expand the list of monitoring options.
3. Click on the plus sign next to RMON in the tree view to expand the list of option links.
4. Click on the Statistics link to display the RMON Statistic table.

The same statistics as those listed for the CLI are displayed.

NOTE: The number of entries in a RMON statistics table directly corresponds to the number of ports on a system. For example, if the system is a 26 port device, there will be 26 entries in the statistics display.

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 be 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

USING THE CLI

A sample RMON history command and its syntax is shown below:

```
BigIron(config)# rmon history 1 interface 1 buckets 10 interval 10 owner nyc02
```

Syntax: rmon history <entry-number> interface <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.

USING THE WEB MANAGEMENT INTERFACE

1. Log on to the device using a valid user name and password for read-only or read-write access. The System configuration dialog is displayed.
2. Click on the plus sign next to Monitor in the tree view to expand the list of monitoring options.
3. Click on the plus sign next to RMON in the tree view to expand the list of option links.
4. Click on the History link to display the RMON History table.

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.

USING THE CLI

A sample CLI alarm entry and its syntax is shown below:

```
BigIron(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>

USING THE WEB MANAGEMENT INTERFACE

This display is not supported on the Web management interface.

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.

USING THE CLI

A sample entry and syntax of the event control table is shown below:

```
BigIron(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>

USING THE WEB MANAGEMENT INTERFACE

This display is not supported on the Web management interface.

NetFlow

NetFlow is a feature that collects information about the traffic that your BigIron Chassis device receives and forwards, and exports that information to external third-party data collectors. NetFlow-compatible accounting and billing applications can use the exported information to create reports, bill customers for network usage, and so on.

The traffic information is collected and exported as flows. A ***flow*** consists of the following information:

- Ingress port (the port that receives the traffic)
- Egress port (the port out which the traffic is forwarded)
- Source IP address
- Destination IP address
- IP Protocol type (IP, ICMP, TCP, UDP, and so on)
- Source TCP or UDP port
- Destination TCP or UDP port
- Type of Service (ToS)
- TCP flags

Traffic collection occurs for the interfaces on which you enable NetFlow.

NetFlow uses the BigIron Layer 4 session table as its flow cache. The session table is active regardless of the state of NetFlow and is used as a fast-path for forwarding IP packets based on Layer 3 and Layer 4 information. By using the session table, NetFlow ensures that the exported data reflects the traffic that has actually been forwarded, and thus excludes traffic that is filtered out without being forwarded.

The VM1 has a separate flow cache (Layer 4 session table) on the VM and on each VSP CPU.

- VM flow cache – This flow cache contains entries for broadcast traffic, multicast traffic, and for traffic addressed to the Foundry device itself. The VM also contains the aggregate caches, if configured.
- VSP CPU flow caches – The flow cache on a given VSP CPU contains entries for traffic that is forwarded on that module.

You can display and change the maximum flow cache size separately for the VM and for each VSP CPU.

NOTE: Other features including ACLs, rate limiting, Policy-based Routing (PBR) and Network Address Translation (NAT) also use the Layer 4 session table. If you change the session table size, the change can affect all these features.

For information about how the forwarding modules are allocated to the VSP CPUs, see the “Using the Velocity Management Module“ chapter in the *Foundry Switch and Router Installation and Basic Configuration Guide*.

Hardware Support

In releases earlier than 07.6.01b, NetFlow requires the Velocity Management Module version 1 (VM1) or higher. The feature also is supported in the NetIron Internet Backbone router, on non-NPA POS modules.

NetFlow is supported on a VM1 in a BigIron Chassis device.In release 07.6.01 and later, NetFlow is supported on the following:

- VM1 in a BigIron or NetIron Chassis device
- POS OC-3 and OC-12 modules in a BigIron or NetIron Chassis device
- ATM modules in a BigIron or NetIron Chassis device

Flow Aging and Export

NetFlow collects and exports flows once they age out of the flow cache (session table). NetFlow ages a flow out of the flow cache (session table) when one of the following occurs:

- The flow is for a TCP session and the session has been terminated normally by a TCP FIN or RST packet.
- The flow is inactive. By default, an inactive flow is one that has not been used within the last 60 seconds as a fast-path to forward a packet. You can change the age for inactive flows to a value from 10 – 600 seconds.
- The flow has been active for a long time. By default, an active flow is aged out after 30 minutes. You can change the age for active flows to a value from 1 – 60 minutes.
- The cache is full and NetFlow needs room for new entries.

By default, NetFlow exports all the aged out flows for all the interfaces on which you enable Flow Switching. You can streamline flow export using the following methods:

- Disable export for specific transport protocols. You can disable export of flows for TCP traffic, flows for UDP traffic, or all flows ***except*** flows or TCP or UDP traffic.
- Configure aggregate caches. An aggregate cache aggregates separate flows into a single flow based on specific information, and exports the aggregated flow instead of the individual flows. See “Aggregate Caches”.

Aggregate Caches

By default, NetFlow creates and exports a separate flow for each unique combination of the flow information. To streamline data export, you can configure additional, aggregate caches, to aggregate flows based on specific data and export the aggregate flows instead of individual flows. The collector receives the aggregate flows instead of the individual flows.

NetFlow supports the following types of aggregate caches. For each cache, the information by which the flows are aggregated is listed.

- Autonomous System (AS)
  - Input and output interfaces
  - Source and destination BGP4 AS
- Source prefix
  - Input interface
  - Source network mask and prefix (for example, 192.168.2.0/24)
  - Source BGP4 AS
- Destination prefix
  - Output interface
  - Destination network mask and prefix
  - Destination BGP4 AS
- Source and destination prefix
  - Input and output interfaces
  - Source and destination network masks and prefixes

    - Source and destination BGP4 AS
- IP protocol and application port
    - Source and destination IP protocol
    - Source and destination TCP or UDP port numbers, if the IP protocol is 6 (TCP) or 17 (UDP)

The aggregate caches are separate from the main flow cache. When a flow is aged out of the flow cache, the flow is moved to the applicable aggregate cache, where it is aggregated with other flows. The aggregate flows are then exported to the collector.

You can configure one or more of the aggregate caches, up to all five of them.

Parameter settings for the main flow cache and aggregate caches are independent. Changing a parameter setting for one cache does not affect the setting for the same parameter in other caches.

NOTE: Aggregate caching does not reduce the amount of NetFlow traffic collected for individual interfaces. Each module locally collects non-aggregated flows, then sends them to the VM1, which aggregates them in the appropriate caches for export.

NOTE: You can configure all the aggregate cache types regardless of whether the device is running Layer 2 Switch software or Layer 3 Switch software. However, some flow information may be inapplicable and will appear as zeros in the exported flows.

Collectors

NetFlow packages the expired flows into UDP packets and sends the packets to external devices called ***collectors***. When you configure a collector on the Foundry device, you specify its IP address and the UDP port number to receive the exported flows.

You can use a total of 15 collectors. You can specify up to 10 collectors for the main flow cache, and one additional collector for each aggregate cache. If you specify more than one collector for the main cache, NetFlow load balances the flows that it exports from the main cache based on the source IP addresses of the flows. Load balancing does not apply to the flows exported from the aggregate caches.

Source Interfaces

By default, the Foundry device uses the interface that is connected to a given collector as the source interface for that collector. You can change the source interface for a collector to one of the following:

- Ethernet or POS port – NetFlow sends the export packets out the specified interface.
- Loopback interface – NetFlow sends the export packets from the specified loopback address, using a physical port connected to the collector to transmit the packet.
- Null interface – NetFlow continues to collect flows but does not export them to the collector. Use this type of interface when you want to administratively stop flow export without stopping flow collection and without removing configuration information.

Export Packet Format Versions

NetFlow places flows into UDP packets and sends the packets to the collector. The format version of the packets can differ depending on the type of information you want to export. The Foundry implementation of NetFlow supports the following format versions:

- Version 1 – Version 1 NetFlow export packets contain basic flow information used by early versions of NetFlow-based applications. The information includes ingress and egress ports, IP protocol, source and destination IP addresses and TCP or UDP ports, and IP ToS values.
- Version 5 – Version 5 contains all the information in Version 1 packets, and adds information for Border Gateway Protocol version 4 (BGP4) AS numbers, as well as flow sequence numbers. This is the default.

- Version 8 – Version 8 supports aggregate flows. When you enable an aggregate cache, the software automatically uses format 8 for export flows from the aggregate cache. Version 1 or 5 is still used for exported flows from the main cache.

Format Version 1

Table A.2 lists the fields in format version 1 NetFlow packets.

Table A.2: NetFlow Format Version 1

Field	Description
Header Fields	
UINT16 version	Current version (1)
UINT16 count	Number of records in this PDU
UINT32 SysUptime	Current time in msecs since the Foundry device was booted
UINT32 unix_secs	Current seconds since 0000 UTC 1970
UINT32 unix_nsecs	Residual nanoseconds since 0000 UTC 1970
Flow Data Fields	
ipaddrtype srcaddr	Source IP Address
ipaddrtype dstaddr	Destination IP Address
ipaddrtype nexthop	Next hop router's IP Address
UINT16 input	Input interface index
UINT16 output	Output interface index
UINT32 dPkts	Packets sent in Duration (milliseconds between first and last packet in this flow)
UINT32 dOctets	Octets sent in Duration (milliseconds between first and last packet in this flow)
UINT32 First	SysUptime at start of flow
UINT32 Last	SysUptime of last packet of the flow
UINT16 srcport	TCP/UDP source port number (FTP, Telnet, and so on)
UINT16 dstport	TCP/UDP destination port number
UINT16 pad	Pad to word boundary
UINT8 prot	IP protocol (6=TCP, 17=UDP, and so on)
UINT8 tos	IP Type of Service (ToS)
UINT8 tcp_flags	Cumulative OR of TCP flags
UINT8 pad	Pad to word boundary
UINT16 pad	Pad to word boundary
UINT32 reserved	Reserved for future use

Format Version 5

Table A.3 lists the fields in format version 5 NetFlow packets.

Table A.3: NetFlow Format Version 5

Field	Description
Header Fields	
UINT16 version	Current version (1)
UINT16 count	Number of records in this PDU
UINT32 SysUptime	Current time in msecs since the Foundry device was booted
UINT32 unix_secs	Current seconds since 0000 UTC 1970
UINT32 unix_nsecs	Residual nanoseconds since 0000 UTC 1970
UINT32 flow_sequence	Sequence number of total flows seen
UINT8 engine_type	Type of flow switching engine
UINT8 engine_id	Slot number of the flow switching engine
UINT16 reserved	Reserved for future use
Flow Data Fields	
ipaddrtype srcaddr	Source IP Address
ipaddrtype dstaddr	Destination IP Address
ipaddrtype nexthop	Next hop router's IP Address
UINT16 input	Input interface index
UINT16 output	Output interface index
UINT32 dPkts	Packets sent in Duration (milliseconds between first and last packet in this flow)
UINT32 dOctets	Octets sent in Duration (milliseconds between first and last packet in this flow)
UINT32 First	SysUptime at start of flow
UINT32 Last	SysUptime of last packet of the flow
UINT16 srcport	TCP/UDP source port number (FTP, Telnet, and so on)
UINT16 dstport	TCP/UDP destination port number
UINT8 pad	Pad to word boundary
UINT8 tcp_flags	Cumulative OR of TCP flags
UINT8 prot	IP protocol (6=TCP, 17=UDP, and so on)
UINT8 tos	IP Type of Service (ToS)
UINT16 dst_as	Destination peer/origin Autonomous System

Table A.3: NetFlow Format Version 5 (Continued)

Field	Description
UINT16 src_as	Source peer/origin Autonomous System
UINT8 dst_mask	Destination route's mask bits
UINT8 src_mask	Source route's mask bits
UINT16 pad	Pad to word boundary

Format Version 8

Table A.4 lists the fields in format version 8 NetFlow packets. Version 8 is used to export aggregated flows. Each of the aggregate cache types has a different format, as shown in the table.

Table A.4: NetFlow Format Version 8

Field	Description
Header Fields	
UINT16 version	Current version (1)
UINT16 count	Number of records in this PDU
UINT32 SysUptime	Current time in msecs since the Foundry device was booted
UINT32 unix_secs	Current seconds since 0000 UTC 1970
UINT32 unix_nsecs	Residual nanoseconds since 0000 UTC 1970
UINT32 flow_sequence	Sequence number of total flows seen
UINT8 engine_type	Type of flow switching engine
UINT8 engine_id	Slot number of the flow switching engine
UINT8 aggregation	Aggregation method being used
UINT8 agg_version	Version of the aggregation export (2)
UINT32 reserved	Reserved for future use
Flow Data Fields for AS Aggregation Flows	
UINT32 flows	Number of flows
UINT32 dPkts	Packets sent in Duration
UINT32 dOctets	Octets sent in Duration
UINT32 First	SysUptime at start of flow
UINT32 Last	SysUptime of last packet of flow
UINT16 src_as	Originating AS of source address
UINT16 dst_as	Originating AS of destination address
UINT16 input	Input interface index

Table A.4: NetFlow Format Version 8 (Continued)

Field	Description
UINT16 output	Output interface index
Flow Data Fields for Protocol Port Flows	
UINT32 flows	Number of flows
UINT32 dPkts	Packets sent in Duration
UINT32 dOctets	Octets sent in Duration
UINT32 First	SysUptime at start of flow
UINT32 Last	SysUptime of last packet of flow
UINT8 prot	IP protocol (6=TCP, 17=UDP, and so on)
UINT8 pad	Pad to word boundary
UINT16 reserved	Reserved for future use
UINT16 srcport	TCP/UDP source port number (FTP, Telnet, and so on)
UINT16 dstport	TCP/UDP destination port number
Flow Data Fields for Source Prefix Flows	
UINT32 flows	Number of flows
UINT32 dPkts	Packets sent in Duration
UINT32 dOctets	Octets sent in Duration
UINT32 First	SysUptime at start of flow
UINT32 Last	SysUptime of last packet of flow
ipaddrtype src_prefix	Source prefix
UINT8 src_mask	Source address prefix mask bits
UINT8 pad	Pad to word boundary
UINT16 src_as	Originating AS of source address
UINT16 input	Input interface index
Flow Data Fields for Destination Prefix Flows	
UINT32 flows	Number of flows
UINT32 dPkts	Packets sent in Duration
UINT32 dOctets	Octets sent in Duration
UINT32 First	SysUptime at start of flow
UINT32 Last	SysUptime of last packet of flow
ipaddrtype dst_prefix	Destination prefix
UINT8 dst_mask	Destination address prefix mask bits
UINT8 pad	Pad to word boundary

Table A.4: NetFlow Format Version 8 (Continued)

Field	Description
UINT16 dst_as	Originating AS of destination address
UINT16 output	Output interface index
Flow Data Fields for Source and Destination Prefix Flows	
UINT32 flows	Number of flows
UINT32 dPkts	Packets sent in Duration
UINT32 dOctets	Octets sent in Duration
UINT32 First	SysUptime at start of flow
UINT32 Last	SysUptime of last packet of flow
ipaddrtype src_prefix	Source prefix
ipaddrtype dst_prefix	Destination prefix
UINT8 dst_mask	Destination address prefix mask bits
UINT8 src_mask	Source address prefix mask bits
UINT16 reserved	Reserved for future use
UINT16 src_as	Originating AS of source address
UINT16 dst_as	Originating AS of destination address
UINT16 input	Input interface index
UINT16 output	Output interface index

Configuring a Chassis Device for NetFlow

To configure NetFlow:

- Enable individual interfaces for Flow Switching. Flows are collected and exported only for the interfaces on which you enable Flow Switching.
- Enable NetFlow globally.
- Specify collector information. The collector is the device to which you are exporting the NetFlow data. You can use up to 10 collectors.

NOTE: If you plan to use aggregate caches instead, you do not need to globally enable NetFlow or specify collector information. Instead, you perform this configuration as part of the aggregate cache configuration.

The following configuration tasks are optional.

- Change the maximum number of main cache entries.
- Specify the interface that will source the exported data. By default, the source interface is the one that is attached through the network to the collector. You can specify an Ethernet port, loopback interface, or null interface as the source for NetFlow export packets.
- Disable export for specific transport protocols.
- Change the NetFlow version for export packets from the main cache. You can specify 1 or 5. The default is 5.

- Enable collection of AS information. AS information collection is disabled by default.
- Set the cache timeout.
- Configure aggregate caches.

Enabling Flow Switching on an Interface

NetFlow exports flow data only for the interfaces on which you enable Flow Switching. You can enable Flow Switching on the following types of interfaces:

- Ethernet interfaces
- POS interfaces
- Virtual routing interfaces (flows are collected and exported for all the ports in the VLAN on which the virtual interface is configured)

To enable Flow Switching on an interface, enter commands such as the following:

```
BigIron(config)# interface ethernet 1/1
BigIron(config-if-1/1)# ip route-cache flow
BigIron(config-if-1/1)# exit
```

Syntax: [no] ip route-cache flow

Enabling NetFlow

To enable NetFlow, enter the following command at the global CONFIG level of the CLI:

```
BigIron(config)# ip flow-export enable
```

Syntax: [no] ip flow-export enable

NOTE: This command enables the feature globally. However, to begin flow collection and export, you must enable collection for individual interfaces. NetFlow collects and exports flows only for the interfaces on which you enable the feature.

Changing the Export Format Version

By default, NetFlow uses format version 5 for exporting flows from the main cache to the external collector. To change the format version, enter a command such as the following at the global CONFIG level of the CLI:

```
BigIron(config)# ip flow-export version 1
```

Syntax: [no] ip flow-export version 1 | 5

For information about the format versions, see “Aggregate Caches” on page A-9.

NOTE: The format for the main cache is not related to the format for aggregate caches. The software automatically uses format 8 for export packets from the aggregate caches.

Specifying the Collector

NetFlow exports flows to external collectors. You can specify up to 10 collectors for the main flow cache. If you specify more than one collector, NetFlow load balances the exported flows among the collectors based on the source IP addresses in the flows.

NOTE: To specify the collector for an aggregate cache, see “Configuring Aggregation” on page A-20. You must specify an aggregate cache’s collector separately, as part of the configuration for the aggregate cache.

To specify a collector, enter a command such as the following at the global CONFIG level of the CLI:

```
BigIron(config)# ip flow-export destination 10.10.10.1 8080 1
```


Syntax: [no] ip flow-export destination <ip-addr> <udp-portnum> [<collector-id>]

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

The <udp-portnum> specifies the UDP port on the collector that listens for the exported flow packets.

The <collector-id> is a number from 1 – 10. This number applies only to the Foundry device and is not related to configuration information on the collector itself.

- If you are specifying more than one collector, make sure you also specify the collector ID and use a different ID for each collector.
- If you are specifying only one collector, you do not need to specify the ID. In this case, the software automatically assigns ID 1 to the collector.

NOTE: If you do not specify the collector ID, the software always uses ID 1. If you already have added a collector whose ID is 1, and you add another collector with ID 1, the software replaces the older collector with the new collector.

Changing the Size of the Main Flow Caches

The main flow cache use the Layer 4 session tables on the VM and the VSPs. These tables have a configurable maximum number of flows they can contain.

The VM1 has a separate Layer 4 session table on the VM and on each VSP CPU.

- VM Layer 4 session table – This table contains entries for broadcast traffic, multicast traffic, and for traffic addressed to the Foundry device itself.
- VSP CPU session tables – The session table on a given VSP CPU contains entries for traffic that is forwarded on that module.

You must change the maximum flow cache sizes separately for the VM and for the VSP CPUs.

For information about how the forwarding modules are allocated to the VSP CPUs, see the “Using the Velocity Management Module“ chapter in the *Foundry Switch and Router Installation and Basic Configuration Guide*.

NOTE: Other features including ACLs, rate limiting, Policy-based Routing (PBR) and Network Address Translation (NAT) also use the Layer 4 session table. If you change the session table size, the change can affect all these features.

NOTE: If you change the maximum number of Layer 4 session entries, you must reload the software to place the change into effect.

Displaying and Changing the Layer 4 Session Table Size on the VM

To display the maximum number of flows the Layer 4 session table on the VM can contain, enter the following command at any level of the CLI:

```
BigIron# show default values
...<some lines omitted>
System Parameters    Default    Maximum    Current
...<some lines omitted>
session-limit        262144     1000000    1024
```

The “session-limit” row in the “System Parameters” section shows the default maximum number of Layer 4 sessions (main cache flows) the VM can have, the maximum currently allowed by the VM, and the maximum you can configure the VM to allow.

To change the maximum number of Layer 4 sessions on the VM, enter a command such as the following at the global CONFIG level of the CLI:

```
BigIron(config)# system-max session-limit 500000
```

Syntax: [no] system-max session-limit <num>

The <num> parameter indicates the maximum number of Layer 4 sessions (main cache flows) the VM can have. You can specify from 1024 – 1000000. The default is 262144.

You must reload the software to place the change into effect. Enter the following commands to reload the software.

```
BigIron(config)# exit
BigIron# reload
```

NOTE: If you plan to also change the size of the VSP Layer 4 session tables now, you can make that change too before reloading the software to place both changes into effect.

Displaying and Changing the Layer 4 Session Table Size on a VSP

To display the maximum number of flows the Layer 4 session table on a VSP can contain, log on to the VSP, then enter the **show usage** command. Here is an example:

```
BigIron# rconsole 2 1
BigIron2/1 # show usage

Avail. Sessions      =    1999997  Total Sessions       =    2000000

BigIron2/1 # rconsole-exit
BigIron#
```

Syntax: show usage

This example shows how to log on to a VSP, display the session table size information, and log off the VSP.

To change the maximum number of Layer 4 sessions on each of the VSPs, enter a command such as the following at the global CONFIG level of the CLI:

```
BigIron(config)# server session-vm-limit 1000000
```

Syntax: [no] server session-vm-limit <num>

The <num> parameter indicates the maximum number of Layer 4 sessions (main cache flows) the VM can have. You can specify from 32768 – 5000000 (five million). The default is 2000000 (two million).

The command applies to all the VSPs.

You must reload the software to place the change into effect. Enter the following commands to reload the software.

```
BigIron(config)# exit
BigIron# reload
```

Setting the Source Interface

By default, the Foundry device uses the port that is connected to a collector as the source interface for flows exported to that collector. You can specify an Ethernet port or POS port, a loopback interface, or the null interface as the source for NetFlow export packets.

- Ethernet or POS port – NetFlow sends the export packets out the specified interface.
- Loopback interface – NetFlow sends the export packets from the specified loopback address, using a physical port connected to the collector to transmit the packet.

- Null interface – NetFlow continues to collect flows but does not export them to the collector. Use this type of interface when you want to administratively stop flow export without stopping flow collection and without removing configuration information.

To specify the source interface, enter a command such as the following:

```
BigIron(config)# ip flow-export source ethernet 1/1
```

This command configures port 1/1 to be the source interface for NetFlow packets. Since the command does not specify the collector ID, NetFlow exports the flows to collector 1.

To specify the collector ID, enter a command such as the following:

```
BigIron(config)# ip flow-export source ethernet 1/1 2
```

This command uses port 1/1 as the source for flows exported to collector 2.

Syntax: [no] ip flow-export source ethernet | pos | loopback <portnum> [<collector-id>]

Syntax: [no] ip flow-export source null [<collector-id>]

The **ethernet | pos | loopback** <portnum> specifies a physical port or loopback interface.

The **null** parameter discards the export packets instead of sending them to a collector. However, NetFlow continues to collect flows.

The <collector-id> specifies the collector. If you do not specify the collector ID, the device assumes you mean collector 1.

Disabling Export of Flows for Some Transport Protocols

By default, NetFlow exports flows for all IP protocols (TCP, UDP, IGRP, OSPF, and so on). To reduce flow exports to the collectors, you can disable export of flows for the following:

- TCP
- UDP
- All IP protocols ***except*** TCP and UDP

To disable export of flows, enter a command such as the following:

```
BigIron(config)# ip flow-export protocol-disable udp
```

Syntax: [no] ip flow-export protocol-disable tcp | udp | other

Enabling AS Flow Information

Export version format 8 contains fields for BGP4 AS information. By default, NetFlow does not collect and export the AS information. To enable collection and export of the AS information, enter the following command at the global CONFIG level of the CLI:

```
BigIron(config)# ip flow-export peer-as
```

Syntax: [no] ip flow-export origin-as | peer-as

The **origin-as | peer-as** parameter specifies the type of AS information you want to enable. You can enable one or the other but not both. By default, neither type of AS information is enabled.

Changing the Cache Timeouts

NetFlow uses the following age timers to age flows out of the cache for export.

- Inactive – The inactive timer ages out a flow after it has been unused for the specified number of seconds.
- Active – The active timer ages out a flow that is in use if the flow has remained in use continuously for the specified number of minutes.

NOTE: In addition to using these timers, NetFlow also ages out normally terminated TCP flows, and ages out flows when the cache becomes full. See “Flow Aging and Export” on page A-9.

NOTE: The main flow cache and the aggregate caches (if you configure them) use separate timeouts. Changing the main flow cache's timeouts does not affect the timeouts for the aggregate caches.

To change a flow aging timer, enter a command such as the following at the global CONFIG level of the CLI:

```
BigIron(config)# ip flow-cache timeout active 45
```

This command changes the maximum age for active flows to 45 minutes.

Syntax: [no] ip flow-cache timeout active <mins> | inactive <secs>

The **active** <mins> parameter specifies the maximum number of minutes an active flow can remain in the cache.

The **inactive** <secs> parameter specifies the maximum number of seconds an inactive flow can remain in the cache.

Configuring Aggregation

By default, NetFlow exports a separate flow for each unique set of flow information. You can consolidate flows by creating aggregate caches. An aggregate cache consolidates individual flows based on specific information in the flows. You can configure the following types of aggregate caches:

- BGP4 AS information
- Source and destination network prefixes
- Source network prefix
- Destination network prefix
- IP protocol and application port number

See "Aggregate Caches" on page A-9 for a list of the flow data fields that are aggregated in each type of aggregate cache.

Each cache requires separate configuration. When you configure an aggregate cache, you can configure the following parameters:

- Collector IP address and UDP port
- Maximum number of flows
- Inactive timeout
- Active timeout

Each aggregate cache has its own settings for these parameters. The parameters are independent from the parameters for the main flow cache. The settings for the main flow cache do not affect the settings for similar parameters in the aggregate caches.

NOTE: NetFlow automatically uses export format version 8 for export packets from an aggregate cache. The format for aggregate flows is unrelated to the format for export packets from the main cache. You do not need to configure the export format for the aggregate caches.

Configuring an Aggregate Cache

To begin configuration of an aggregate cache, enter a command such as the following at the global CONFIG level of the CLI:

```
BigIron(config)# ip flow-aggregation cache as
BigIron(config-flow-cache_as)#
```


This command begins configuration for the AS aggregate cache. Notice that the CLI prompt changes to the configuration level for the aggregate cache. At the aggregate cache's configuration level, you can change cache parameters. Here is an example:

```
BigIron(config)# ip flow-aggregation cache as
BigIron(config-flow-cache_as)# cache entries 2046
BigIron(config-flow-cache_as)# cache timeout inactive 200
BigIron(config-flow-cache_as)# cache timeout active 45
BigIron(config-flow-cache_as)# export destination 10.42.42.1 9992
BigIron(config-flow-cache_as)# enabled
```

The **cache** commands change cache parameters. The **enable** command enables the cache. A cache does not go into effect until you enable it.

Command Syntax

The following command begins configuration of an aggregate cache.

Syntax: [no] ip flow-aggregation cache as | destination-prefix | prefix | protocol-port | source-prefix

The **as | destination-prefix | prefix | protocol-port | source-prefix** parameter specifies the cache type.

- **as** – Configures an AS cache. Flows are aggregated based on AS number.
- **destination-prefix** – Configures a destination prefix cache. Flows are aggregated based on destination network prefix.
- **prefix** – Configures a prefix cache. Flows are aggregated based on both source and destination network prefixes.
- **protocol-port** – Configures a protocol port cache. Flows are aggregated based on source and destination IP protocol port.
- **source-prefix** – Configures a source prefix cache. Flows are aggregated based on source network prefix.

The following command specifies the collector. You can specify one collector for an aggregate cache.

Syntax: [no] export destination <ip-addr> <udp-portnum>

The following command specifies the maximum number of flows the cache can contain.

Syntax: [no] cache entries <num>

The <num> parameter specifies the maximum number of flows and can be from 1024 – 524288. The default is 4096.

The following commands specify the active and inactive timeouts.

Syntax: [no] cache timeout inactive <secs>

Syntax: [no] cache timeout active <mins>

The **inactive** <secs> parameter specifies the maximum number of seconds an inactive flow can remain in the cache.

The **active** <mins> parameter specifies the maximum number of minutes an active flow can remain in the cache.

The following command enables the cache.

Syntax: [no] enabled

NOTE: The **enabled** command is required to enable the cache. The **cache** commands are optional.

Displaying NetFlow Information

You can display the following information:

- Export configuration information and statistics

- Flow cache configuration information and the flows in the caches

NOTE: The CLI has different commands for displaying the VM flow cache and a VSP's flow cache.

- Statistics for aggregate caches

Displaying Export Information

To display configuration information and statistics for NetFlow export, enter the following command at any level of the CLI:

```
BigIron(config)# show ip flow export
Flow export is enabled
Version 5 flow records
Autonomous System information not included in export datagrams
COLLECTOR 1:
  Exporting using source interface Ethernet 3/1
  Collector IP address 10.1.1.1
  Collector UDP port 3065

COLLECTOR 2:
  Exporting using source interface Ethernet 3/2
  Collector IP address 10.1.1.2
  Collector UDP port 3066

COLLECTOR 3:
  Exporting using source interface Ethernet 3/3
  Collector IP address 20.1.2.3
  Collector UDP port 3000

Cache for protocol-port aggregation:
Aggregation flow export is enabled
Exporting using source interface Ethernet 3/5
Exporting flows to  10.2.2.2 (3233)

Cache for destination-prefix aggregation:
Aggregation flow export is enabled
Exporting using source interface Ethernet 3/4
Exporting flows to  10.2.2.2 (3235)

20000 flows exported in 15800 udp datagrams
0 flows failed to export
0 export packets were dropped
```

Syntax: show ip flow export

This command shows the following information.

Table A.5: Export Configuration and Statistics

This Line...	Displays...
Main flow cache information	
Flow export	The state of the feature, which can be one of the following: • disabled • enabled
Version	The version of the flow UDP packets sent to the collector. The version can be one of the following: • 1 • 5 (the default) If peer AS or origin AS information is included in the flows, this is indicated by "peer-as" or "origin-as" following the version information.
Autonomous system information	Whether you have enabled collection of AS information.
COLLECTOR	The collectors you have configured for exports from the main flow cache. The non-aggregated flows go to these collectors. The following information is listed for each collector: • Source interface • Collector IP address • Collector UDP port
Aggregate cache information	
Cache type	The aggregate cache type.
Cache state	Whether the cache is enabled or disabled.
Source interface	The source interface for exporting flows from the cache.
Collector	The IP address and UDP port of the collector.
Export statistics **Note**: These fields apply to the flows exported from the main cache and from the aggregate caches.	
Flows exported	The number of flows exported to the collector and the number of UDP datagrams used to send the flows.
Flows failed to export	The number of flows that could not be exported. This can occur if the collector is not reachable.
Export packets dropped	The number of flows that were dropped rather than sent to the collector. This can occur if too many flows are expiring (becoming eligible for export) at the same time and there is not enough buffer to hold all of them.

Displaying the Main Flow Cache on the VM

To display the main flow cache on the VM, enter the following command at any level of the CLI:

```
BigIron(config)# show ip cache flow
Flow Cache Active Timeout is 60 minutes
Flow Cache Inactive Timeout is 60 seconds

Flow Cache is Active for the following interfaces:
ethernet 3/1
ethernet 4/1

Input If     Src Prefix  Output If     Dst Prefix    Packets    Octets
     129       20.1.1.1        193       10.1.1.1         10       650
     129       20.1.1.1        193       10.1.1.1          8       600
     193       10.1.1.1        129       20.1.1.1         10       688
     193       10.1.1.1        129       20.1.1.1          7       602
```

Syntax: show ip cache flow

This command shows the following information.

Table A.6: Flow Cache Information

This Field...	Displays...
Inactive timeout	The maximum number of seconds an inactive flow can remain in the main cache.
Active timeout	The maximum number of minutes an active flow can remain in the main cache.
Flow Cache is Active for the following interfaces	The interfaces on which NetFlow Caching is enabled.
Input If	The SNMP number of the interface on which the Foundry device received traffic for the flow.
Src Prefix	The network prefix of the source network.
Output If	The SNMP number of the interface on which the Foundry device transmitted traffic for the flow.
Dst Prefix	The network prefix of the destination network.
Packets	The number of packets that have been forwarded using this flow entry.
Octets	The number of octets that have been forwarded using this flow entry.

Displaying the Flow Cache on a VSP

To display the flow cache on a VSP:

- Determine the VSP that contains the cache you want to display. Each forwarding module is managed by a specific VSP.
- Log on to the VSP.
- Display the flow cache on the VSP.

Here is an example. The following command lists the forwarding modules that each VSP is managing.

```
BigIron# show vm-map
slot 3 (weight 80 x 100M) will be processed by VM 2/1 (weight 1)
slot 4 (weight 24 x 100M) is processed by VM 2/2 (weight 24)
```

The following commands log on to a specific VSP and display the flow cache on the VSP.

```
BigIron# rconsole 2 1
BigIron2/1 # enable
BigIron2/1 # show ip cache flow 0
Flow Cache Active Timeout is 60 minutes
Flow Cache Inactive Timeout is 10 seconds

Index Input If     Src Prefix  Output If     Dst Prefix    Packets    Bytes
    0      129       20.1.1.1        193       10.1.1.1          4      240
```

Syntax: show ip cache flow <num>

The <num> parameter specifies the entry number at which to start the display. The VSP display does not support paging.

The following command logs out of the VSP and returns the CLI to the VM.

```
BigIron2/1 # rconsole-exit
BigIron#
```

For information about logging on to a VSP, see the “Using the Velocity Management Module“ chapter in the *Foundry Switch and Router Installation and Basic Configuration Guide*.

Displaying an Aggregate Cache

To display statistics and entries for an aggregate cache, enter a command such as the following at any level of the CLI:

```
BigIron(config)# show ip cache flow aggregation as
IP Flow Aggregation Switching Cache, 2883584 bytes
1 active, 65535 inactive
70 ager polls, 0 flow alloc failures (cache space overflow)
Active Timeout 60 minutes, Inactive Timeout 60 seconds

Src AS  Dst AS  Input If  Output If     Flows     Pkts     Octets
    20       10       193        129         2       17       1290
    10       20       129        193         2       18       1250
```

This example shows information for the AS aggregate cache.

Syntax: show ip cache flow aggregation as | destination-prefix | prefix | protocol-port | source-prefix

Here are examples of the output for the other types of aggregate caches. The following example shows the destination-prefix aggregate cache.

```
BigIron(config)# show ip cache flow aggregation destination-prefix
IP Flow Aggregation Switching Cache, 2883584 bytes
3 active, 65533 inactive
65 ager polls, 0 flow alloc failures (cache space overflow)
Active Timeout 60 minutes, Inactive Timeout 60 seconds

Output If     Dst Prefix  Msk    AS     Flows     Pkts     Octets
      193       10.1.1.1   24    20         2       18       1250
      129       20.1.1.1   24    10         2       17       1290
```

The following example shows the prefix aggregate cache.

```
BigIron(config)# show ip cache flow aggregation prefix
IP Flow Aggregation Switching Cache, 2883584 bytes
3 active, 65533 inactive
88 ager polls, 0 flow alloc failures (cache space overflow)
Active Timeout 60 minutes, Inactive Timeout 60 seconds

Input If     Src Prefix  Msk  Output If     Dst Prefix  Msk     Flows    Pkts
     129       20.1.1.1   24        193       10.1.1.1   24         2      18
     193       10.1.1.1   24        129       20.1.1.1   24         2      17
```

The following example shows the protocol-port aggregate cache.

```
BigIron(config)# show ip cache flow aggregation protocol-port
IP Flow Aggregation Switching Cache, 2883584 bytes
3 active, 65533 inactive
97 ager polls, 0 flow alloc failures (cache space overflow)
Active Timeout 60 minutes, Inactive Timeout 60 seconds

Protocol  Source Port  Dest Port     Flows     Pkts     Octets
       6         1360         23         1       10        688
       6           23       1360         1        8        600
       6         1361         23         1        7        602
       6           23       1361         1       10        650
```

The following example shows the source-prefix aggregate cache.

```
BigIron(config)# show ip cache flow aggregation source-prefix
IP Flow Aggregation Switching Cache, 2883584 bytes
3 active, 65533 inactive
59 ager polls, 0 flow alloc failures (cache space overflow)
Active Timeout 60 minutes, Inactive Timeout 60 seconds

Input If     Src Prefix  Msk    AS     Flows     Pkts     Octets
     193       10.1.1.1   24    20         2       17       1290
     129       20.1.1.1   24    10         2       18       1250
```


Most of the fields in these displays are the same for all the aggregate cache types, while a few fields are particular to each cache type. The **show ip cache flow aggregation** commands show the following information.

Table A.7: Flow Cache Information

This Field...	Displays...
Common Fields These fields appear for all the options.	
bytes	The number of bytes of flow data in the cache.
active	The number of flows that are in the active state.
inactive	The number of flows that are in the inactive state.
ager polls	The number of times NetFlow has polled the cache for aged out (expired) flows to export to the aggregate cache's collector.
flow alloc failures	The number of times NetFlow was unable to place a new flow into the cache because there was not enough space in the cache for the flow.
Active Timeout	The maximum number of minutes an active flow can remain in the cache.
Inactive Timeout	The maximum number of seconds an inactive flow can remain in the cache.
Input If	The SNMP number of the interface on which the Foundry device received traffic for the flow.
Output If	The SNMP number of the interface on which the Foundry device transmitted traffic for the flow.
Flows	The number of flows that are aggregated into this single flow.
Pkts	The number of packets aggregated in the flow.
Octets	The number of octets aggregated in the flow. **Note**: The display for the **prefix** option does not contain this field.
as Option – Additional Fields	
Src AS	The source AS.
Dst AS	The destination AS.
destination-prefix Option – Additional Fields	
Dst Prefix, Msk	The network prefix and prefix length (mask) for the flow destination.
AS	The destination AS.
prefix Option – Additional Fields	
Src Prefix, Msk	The network prefix and prefix length (mask) for the flow source.
Dst Prefix, Msk	The network prefix and prefix length (mask) for the flow destination.
protocol-port Option – Additional Fields	
Protocol	The IP protocol number.

Table A.7: Flow Cache Information (Continued)

This Field...	Displays...
Source Port	The source TCP or UDP protocol port number, if the IP protocol is 6 (TCP) or 17 UDP.
Dest Port	The destination TCP or UDP protocol port number, if the IP protocol is 6 (TCP) or 17 UDP.
source-prefix Option – Additional Fields	
Src Prefix, Msk	The network prefix and prefix length (mask) for the flow source.
AS	The source AS.

SNMP Support

Foundry MIB version MIB07300.mib contains new objects that enable you to use SNMP to configure and manage the NetFlow feature on a BigIron Chassis device.

Table A.8 lists the NetFlow objects in the Foundry MIB.

Table A.8: Foundry NetFlow Objects

Object	Description
snNetFlowGblEnable	The state of the NetFlow export feature. The state can be one of the following: • 0 – disabled • 1 – enabled
snNetFlowGblVersion	The export version, which can be one of the following: • 1 • 5 (the default) • 8
snNetFlowGblProtocolDisable	The transport layer protocols for which flow export is disabled. • 0 – Export of all flows ***except*** TCP and UDP is disabled. • 1 – Export of all TCP flows is disabled. • 2 – Export of all UDP flows is disabled.
snNetFlowGblActiveTimeout	The maximum number of minutes an active flow can remain in the NetFlow cache. The object can have a value from 1 – 60 minutes. The default is 60 minutes.
snNetFlowGblInactiveTimout	The maximum number of seconds an inactive flow can remain in the NetFlow cache. The object can have a value from 10 – 600 seconds. The default is 60 seconds.
snNetFlowCollectorTable	Contains NetFlow collector objects. See “snNetFlowCollectorTable” on page A-29.
snNetFlowAggregationTable	Contains NetFlow aggregation objects. See “snNetFlowAggregationTable” on page A-29.

snNetFlowCollectorTable

This table contains the following NetFlow configuration objects.

Table A.9: snNetFlowCollectorTable Objects

Object	Description
snNetFlowCollectorEntry	An entry in the table. Each entry contains the following objects.
snNetFlowCollectorIndex	The table's index.
snNetFlowCollectorIp	The IP address of the NetFlow collector.
snNetFlowCollectorUdpPort	The NetFlow UDP port on the collector.
snNetFlowCollectorSourceInterface	The source port for NetFlow packets exported to the collector. The value can be one of the following: • A valid port number – This is the port you configured to be the source port. • 0 – No source port has been specified. In this case, the software uses the port that is connected to the collector as the source port.
snNetFlowCollectorRowStatus	The configuration status of this entry in the table. The configuration status can be one of the following: • 1 – other • 2 – valid • 3 – delete • 4 – create

snNetFlowAggregationTable

This table contains the following NetFlow configuration objects.

Table A.10: snNetFlowAggregationTable Objects

Object	Description
snNetFlowAggregationEntry	An entry in the table. Each entry contains the following objects.
snNetFlowAggregationIndex	The table's index. The index indicates the aggregation scheme for this table row and can be one of the following: • 1 – AS aggregation • 2 – Protocol-port aggregation • 3 – Destination prefix aggregation • 4 – Source prefix aggregation • 5 – Prefix aggregation The remaining table entries apply to the specified aggregation scheme.
snNetFlowAggregationIp	The IP address of the NetFlow collector.

Table A.10: snNetFlowAggregationTable Objects (Continued)

Object	Description
snNetFlowAggregationUdpPort	The NetFlow UDP port of the collector.
snNetFlowAggregationSourceInterface	The source port for NetFlow packets exported to the collector. The value can be one of the following: • A valid port number – This is the port you configured to be the source port. • 0 – No source port has been specified. In this case, the software uses the port that is connected to the collector as the source port.
snNetFlowAggregationNumberOfCacheEntries	The maximum number of aggregated flows the aggregation cache can contain.
snNetFlowAggregationActiveTimeout	The maximum number of minutes an active flow can remain in the NetFlow aggregation cache. The object can have a value from 1 – 60 minutes. The default is 30 minutes.
snNetFlowAggregationInactiveTimeout	The maximum number of seconds an inactive flow can remain in the NetFlow aggregation cache. The object can have a value from 10 – 600 seconds. The default is 15 seconds.
snNetFlowAggregationEnable	The state of the aggregation scheme. The state can be one of the following: • 0 – disabled • 1 – enabled
snNetFlowAggregationRowStatus	The configuration status of this entry in the table. The configuration status can be one of the following: • 1 – other • 2 – valid • 3 – delete • 4 – create

sFlow

NOTE: The CLI syntax and configuration procedures in release 07.6.00 or later are different from those in earlier releases. If you are migrating from a configuration saved while running an earlier release, you will need to reconfigure sFlow using the information in this section.

NOTE: If you are using sFlow in a release earlier than 07.6.01, see the release notes for your release for sFlow information.

NOTE: On BigIron MG8 and NetIron 40G, sFlow and port mirroring can be configured simultaneously on an individual port starting with software release 02.1.00 and later.

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.

Configuration Considerations

Hardware Support

sFlow is supported on the following hardware:

- Chassis device managed by a JetCore management module. The feature is supported for all Ethernet ports (10/100, Gigabit, and 10 Gigabit) and for POS OC-3c and OC-12c ports.
- Chassis device managed by a VM1. The feature is supported for 10/100 and Gigabit Ethernet ports and for POS OC-3c and OC-12c ports.
- 10 Gigabit Ethernet ports managed by an IronCore or JetCore management module.
- POS OC-3c and OC-12c ports, regardless of the IronCore or JetCore management module that is managing the Chassis device.

NOTE: You can use a 10 Gigabit Ethernet module or a POS OC-3c or OC-12c module with a JetCore or IronCore management module. sFlow is supported on the module in either case. If you are using a JetCore management module, the JetCore information in this section applies. If you are using an IronCore management module, the IronCore information applies.

- FastIron 4802. The feature is supported on all ports.
- FastIron Edge Switches. FES devices support sFlow packet sampling for inbound and outbound traffic on sFlow-enabled ports.

On these devices, sample data is collected from inbound traffic on ports enabled for sFlow. Outbound traffic is sampled on the FastIron Edge Switches only. However, both traffic directions are counted for byte and packet counter statistics sent to the collector.

sFlow ***is not*** supported on the following hardware:

- IronCore Chassis devices managed by an IronCore management module other than the VM1 module
- OC-48c POS ports (either NPA or non-NPA) or ATM ports, regardless of the management module type

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 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 A-36 and “Displaying sFlow Information” on page A-38.

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:

- For a VM1, all traffic received on an IronCore Ethernet port enabled for sFlow is sent to the CPU for processing. This occurs regardless of the sampling rate. JetCore Ethernet ports send only the sampled traffic to the CPU. Enable sFlow on an IronCore port only if you are certain you want to sample the data from that port.
- For a JetCore management module, when sFlow is enabled on a JetCore Ethernet port, sampling is enabled on the IPC that manages the port. You can enable sFlow on all the ports managed by the same IPC with minimal additional impact on performance.
- Changing the sampling rate does not affect sFlow performance for Ethernet ports managed by a VM1 but does affect performance for ports managed by JetCore. For ports managed by JetCore, a high sampling rate can reduce performance. Use a low sampling rate for better performance.

Port Monitoring

Port monitoring and sFlow are not supported together. If you enable port monitoring on any port, sFlow is disabled for all ports.

sflow Support for IPv6 Packets

Foundry’s implementation of sFlow features 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.

Sampling of IPv6 Packets

Sampling of IPv6 packets on the VM1 is performed by the TSPs for the port that has sFlow enabled. A sampled packet is sent to the management processor; however, if IPv6 packets are sent to the management processor before it is sampled, the management processor samples the packet.

On the NetIron 4802, IPv6 sampling is done by the IPC or IGC. The DMA uses the sampling rate setting to selectively mark the monitoring bit in an incoming packet's header. Marked packets tell the CPU that the packets are subject to sFlow sampling.

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:

```
BigIron(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 A-31.

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:

```
BigIron(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. In software releases prior to 07.6.03, all sFlow-enabled ports use the default sampling rate, which is 512. With a sampling rate of 512, on average, one in every 512 packets forwarded on an interface is sampled.

Beginning with software release 07.6.03, the default sampling rate depends on the device being configured. The following are the new default sFlow sampling rates:

- Stackables: 128
- 4-slot chassis: 2048
- 8-slot chassis: 8192
- 15-slot chassis: 8192

NOTE: sFlow uses CPU resources to send sFlow samples to the collector. If you set a low sampling rate (such as 32 in a 15-slot chassis with 232 Gigabit Ethernet ports), CPU utilization can become high.

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.

NOTE: On the FastIron Edge Switch, the configured sampling rate and the actual rate are the same. The FES software does not adjust the configured sampling rate.

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.

NOTE: Module sampling rates are not supported on the BigIron MG8.

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 A-38.

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:

```
BigIron(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
- 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.

NOTE: On the FastIron Edge Switch, the configured sampling rate and the actual rate are the same. The FES software does not adjust the configured sampling rate.

NOTE: The FastIron Edge Switch is not limited to the specific sampling rates discussed above as on other Foundry devices.

Prior to software release 07.6.03, the default is 512 packets. Beginning with software release 07.6.03, the default sampling rate depends on the device being configured. The following are the new default sFlow sampling rates:

- Stackables: 128
- 4-slot chassis: 2048
- 8-slot chassis: 8192
- 15-slot chassis: 8192

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:

```
BigIron(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".

NOTE: On the FastIron Edge Switch, the configured sampling rate and the actual rate are the same. The FES software does not adjust the configured sampling rate.

Enabling sFlow Forwarding

sFlow exports data only for the interfaces on which you enable sFlow forwarding. You can enable sFlow forwarding on the following types of interfaces:

- Ethernet interfaces
- POS OC-3c or OC-12c 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 A-31 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 *Foundry Security Guide*.

Enabling sFlow Forwarding

To enable sFlow forwarding, enter commands such as the following:

```
BigIron(config)# sflow enable
BigIron(config)# interface ethernet 1/1 to 1/8
BigIron(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:

```
BigIron(config)# 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 A.11: 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 A-31.
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.

AS Path Information in Sflow

On the BigIron MG8 and NetIron 40G the full AS Path information is available in sFlow on devices running software release 02.2.01 and later.

Clearing sFlow Statistics

To clear the UDP packet and sFlow sample counters in the **show sflow** display, enter the following command:

```
BigIron(config)# 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.

Enhancements to sFlow for MPLS Support

In addition to the regular L2/L3 information export supported across Foundry devices, the NetIron IMR 640 supports exporting of MPLS/VPN information in sFlow when sFlow sampling is configured on VPN endpoint interfaces. This includes VLL, VPLS, and VRF customer endpoint interfaces. This functionality allows service providers to collect sFlow information from VPN customers.

For incoming packets to an endpoint interface sampled by sFlow, the following additional information is collected and exported in the sFlow packets:

- MPLS VC information: including VC name, VC index, and VC label COS;
- MPLS tunnel information: including LSP tunnel name, tunnel index as assigned by the router, and tunnel COS used.

Enhancements to support MPLS in sFlow packet export format are described in “sFlow Version 5"which is available at www.sflow.org.

NOTE: sFlow sampling on MPLS uplink interfaces is not fully supported currently.

Support for sFlow Version 5 on Enterprise Software Releases

Devices running Enterprise software release 08.0.00 and later support sFlow version 5. On these devices, sFlow version 5 modifies and enhances the format of the data sent to the sFlow collector. The new version defines a new datagram syntax for the sFlow agent to report flow samples and interface counters to sFlow collectors, as well as a number of new features.

Starting in release 08.0.00, the sFlow agent exports sFlow version 2 flow samples by default, but you can now export data in either sFlow version 5 or version 2 format. sFlow version 5 collectors are compatible with both sFlow version 2 and version 5 agents running inside a Foundry device.

Release 08.0.00 includes the following features related to sFlow version 5:

- Support for the sFlow version 5 datagram
- Support for the sFlow version 5 MIB
- Support for sub-agents
- Configurable sFlow export packet size
- Support for the new data field and sample type length in flow samples
- Configurable interval for exporting Foundry-specific data structure

New commands have been added to the CLI to allow you to do the following:

- Specify the sFlow version used for exporting sFlow data
- Specify the maximum flow sample size
- Export CPU and memory usage information to the sFlow collector
- Specify the polling interval for exporting CPU and memory usage information to the sFlow collector

Specifying the Version Used for Exporting sFlow Data

In release 08.0.00, by default the sFlow agent on the Foundry device exports sFlow data in version 2 format. You can optionally change this setting so that the sFlow agent on the Foundry device exports data in version 5 format.

To do this, enter the following command:

```
BigIron(config)# sflow version 5
```

Syntax: [no] sflow version 2 | 5

The default is 2.

Specifying the Maximum Flow Sample Size

You can specify the maximum size in bytes of the flow sample sent to the sFlow collector. If a packet is larger than the specified maximum size, then only the contents of the packet up to the specified maximum number of bytes is exported. If the size of the packet is smaller than the specified maximum, then the entire packet is exported.

To specify 1024 bytes as the maximum flow sample size, enter the following command:

```
BigIron(config)# sflow max-packet-size 1024
```

Syntax: [no] sflow max-packet-size <size>

For both sFlow version 2 and version 5, the default maximum flow sample size is 256 bytes.

For sFlow version 5, the maximum flow sample size is 1300 bytes (1000 bytes for devices managed by the VM1).

Exporting CPU and Memory Usage Information to the sFlow Collector

In this release, you can optionally configure the sFlow agent on the Foundry device to export information about CPU and memory usage to the sFlow collector.

To export CPU usage and memory usage information, enter the following command:

```
BigIron(config)# sflow export system-info
```

Syntax: [no] sflow export system-info

By default, CPU usage information and memory usage information are not exported.

Specifying the Polling Interval for Exporting CPU and Memory Usage Information to the sFlow Collector

The polling interval defines how often sFlow data for a port is sent to the sFlow collector. You can optionally set the polling interval used for exporting CPU and memory usage information.

For example, to set the polling interval for exporting CPU and memory usage information to 30 seconds, enter the following command:

```
BigIron(config)# sflow export system-info 30
```

Syntax: [no] sflow export system-info <seconds>

You can specify a polling interval from 5 seconds to 1,800 seconds (30 minutes). The default polling interval for exporting CPU and memory usage information is 300 seconds (5 minutes).

Exporting CPU-Directed Data to the sFlow Collector

In release 08.0.00, you can select which and how often data destined to the CPU (for example, Telnet sessions) is sent to the sFlow collector.

New commands have been added to the CLI to allow you to do the following:

- Enable the sFlow agent to export CPU-directed data
- Specify the sampling rate for exported CPU-directed data

Enabling the sFlow Agent to Export CPU-Directed Data

To enable the sFlow agent on a Foundry device to export data destined to the CPU to the sFlow collector, enter the following command:

```
BigIron(config)# sflow export cpu-traffic
```


Syntax: [no] sflow export cpu-traffic

By default, this command is disabled. The sFlow agent does not send data destined to the CPU to the sFlow collector.

Specifying the Sampling Rate for Exported CPU-Directed Data

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 optionally set the sampling rate for CPU-directed data exported to the sFlow collector. For example, to set this sampling rate to 2048, enter the following command:

```
BigIron(config)# sflow export cpu-traffic 2048
```

Syntax: [no] sflow export cpu-traffic <rate>

The default sampling rate depends on the Foundry device being configured. See “Changing the Sampling Rate” in the “sFlow” section of the *Foundry Enterprise Configuration and Management Guide* for the default sampling rate for each kind of Foundry device.

Appendix B
Policies and Filters

NOTE: This appendix does not describe Access Control Lists (ACLs) or IPX SAP ACLs, which are additional methods for filtering packets. See “Access Control List” on page 6-1, “Configuring BGP4” on page 16-1, and “Configuring IPX SAP Access Control Lists (ACLs)” on page 22-9.

Foundry devices provide a robust array of policies and filters. You can configure policies and filters to do the following:

- Change Quality-of-Service priorities for individual ports, VLANs, Layer 4 flows, static MAC entries, and AppleTalk sockets.
- Configure protocol-based VLANs, IP subnet VLANs, and IPX network VLANs within standard 802.1d port-based VLANs.
- Forward or drop IP packets based on source and destination IP addresses, Layer 4 information (such as TCP or UDP port), or both.
- Learn or drop RIP routes on incoming traffic, based on network address or the RIP neighbor’s IP address.
- Control learning and advertisement of RIP routes, based on network address or the RIP neighbor’s IP address.
- Forward or drop IPX packets based on source and destination network address and socket information.
- Control learning and advertisement of IPX RIP routes.
- Permit or deny access to IPX servers.
- Permit or deny AppleTalk zone and network information to reach other zones.
- Control learning and advertisement of routes learned from BGP4 neighbors. You can filter based on network address information, AS-path information, and community names.
- Redistribute routes among RIP, OSPF, and BGP4.
- In ServerIron Transparent Cache Switching (TCS) configurations, redirect HTTP traffic to cache servers or send the traffic to the Internet.
- In router acceleration configurations, redirect IP or IPX packets received from specific hosts to routers for conventional forwarding instead of directly switching the packets at Layer 3.
- Filter on specific MAC addresses, on Layer 2 multicast packets, and on Layer 2 broadcast packets.

NOTE: Foundry recommends that you use ACLs to handle L4 prioritization on all other platforms, except on NetIron Stackable devices, FastIron Iand II.

This appendix describes the various types of Foundry policies and filters. For each type of policy or filter, the CLI command syntax and the Web management links for configuring the policy or filter are provided. This appendix also refers you to specific configuration procedures.

Scope

Some policies and filters are configured and apply globally, while others are configured globally but apply to individual ports. The following table lists the scope for each type of policy and filter.

Table B.1: Scopes of Policies and Filters

Policy or Filter Type	Scope
QoS policy	Configured and applied to one of the following: • Ports • VLANs • Static MAC entries • Layer 4 sessions • AppleTalk sockets
Cache server redirection policy (applies only to ServerIron's Transparent Cache Switching)	Configured and applied globally or locally: • If configured globally, it is automatically applied to all ports • If configured locally, you must explicitly apply it to each port
Access policy (see forwarding filters)	See Forwarding filters
Forwarding filters • MAC forwarding filters • IP forwarding filters (same as IP access policy) • IPX forwarding filters • TCP/UDP forwarding filters	Configured globally, then applied locally to a port's inbound or outbound policy or filter group. You can use the same policy or filter in a port's inbound policy or filter group and outbound policy or filter group. You also can use the same policy or filter on multiple ports.
Address-lock filter	Configured and applied on individual ports.
Route filters • RIP route filters • IPX RIP route filters • IPX SAP service filters	Configured globally and applied to individual ports
RIP neighbor filters	Configured and applied globally
AppleTalk zone and network filters	Configured and applied on individual ports.
BGP4 filters • BGP4 address • BGP4 AS-path • BGP4 community	Configured and applied globally and in route maps

Table B.1: Scopes of Policies and Filters (Continued)

Policy or Filter Type	Scope
Router acceleration policy • IP switching filter • IPX switching filter	Configured globally, then applied either globally (on Layer 2 Switches) or to individual ports (Layer 3 Switches). The filters are applied to an inbound or outbound policy group. You can use the same policy in an inbound policy group and outbound policy group. For Layer 3 Switches, you also can use the same policy on multiple ports.
Route redistribution filters • RIP • OSPF • BGP4	Configured and applied globally

Default Filter Actions

By default, no policies or filters are defined on Foundry devices. The following table lists the default action when no policy or filter is configured and the default action after you configure a policy or filter. For some types of policies and filters, the default action changes once you configure a policy or filter, regardless of the policy or filter's contents.

Table B.2: Default Policy and Filter Actions

Policy or Filter Type	Default action when no policies or filters are configured	Default action after a policy or filter is configured
QoS policy	Queue all packets in normal or 0 priority queue	Queue all packets in normal or 0 priority queue unless explicitly configured for a higher queue
Cache server redirection policy (applies only to ServerIron's Transparent Cache Switching)	Deny all HTTP packets (do not redirect to cache server)	If global, redirect all HTTP packets; if local, deny (do not redirect) all HTTP packets except for ports to which TCS policy is applied
Access policy (see Forwarding filters)	See Forwarding filters	See Forwarding filters
Forwarding filters • MAC forwarding filters • IP forwarding filters (same as IP access policy) • IPX forwarding filters • TCP/UDP forwarding filters	Permit (forward) all packets	Deny (drop) all packets **Note**: The default action for AppleTalk zone and network filters is always permit. To deny all but specific zones, create permit filters for those zones, then create a deny filter and use the "additional zones" value with the filter.
Address-lock filter	Permit (forward) all packets	Permit only those packets whose source MAC addresses have been learned on the port; drop all others

Table B.2: Default Policy and Filter Actions (Continued)

Policy or Filter Type	Default action when no policies or filters are configured	Default action after a policy or filter is configured
Route filters • RIP neighbor filters • IPX RIP route filters • IPX SAP service filters • AppleTalk zone and network filters • BGP4 address filters • BGP4 AS-path filters • BGP4 community filters	Permit (learn and advertise) all routes or services	Deny (do not learn or advertise) all routes or services
Route filters • RIP route filters	Permit (learn and advertise) all routes or services	Permit (learn and advertise) all routes or services
Router acceleration policy • IP switching filter • IPX switching filter	Permit (accelerate) all packets, thus bypassing router	Deny acceleration to all packets, thus sending packets to router
Route redistribution filter • RIP • OSPF • BGP4	Redistribute all routes if redistribution is enabled.	Once redistribution is enabled, redistribute routes of the specified type unless explicitly denied by filter **Note**: For RIP and OSPF, you must explicitly enable redistribution. Redistribution is enabled by default in BGP4.
Layer 2 broadcast and multicast filters	Allow outbound broadcasts and multicasts on the specified ports	Drop outbound broadcasts or multicasts on the specified ports

Policy and Filter Precedence

QoS

You can apply QoS policies to individual ports, VLANs, static MAC address, Layer 4 sessions, and AppleTalk sockets. If a port is a member of two or more of these items and has different priorities, the priorities are merged. However, the resulting priority is never lower than the highest priority.

Precedence Among Filters on Different Layers

Generally, the Foundry device applies only the type of filter that applies to the traffic. For example, if a packet is a Layer 2 switched packet, then the device evaluates the packet against the port's MAC filters. If a packet is a routed IP packet, the device evaluates the packet against the port's IP access policies.

Foundry recommends that you do not use filters at different layers on the same port. For example, do not use MAC filters and IP access policies on the same port.

NOTE: You cannot use Layer 2 filters to filter for Layer 4 information. To filter for Layer 4 information, use IP access policies (filters).

NOTE: If you do choose to apply filters for multiple layers to the same port, note that Layer 2 MAC filters can affect the Layer 3 IP traffic that a port permits or denies on multinetted interfaces. A multinetted interface has multiple IP subnet interfaces on the same port. MAC filters can filter on the Ethertype field. This field includes Layer 3 protocol information and identifies packets as IP packets, ARP packets, and so on.

If you configure a MAC filter, then leave the default action as “deny any”, all packets from one of the IP subnet addresses to another address on the same multinetted interface that do not match the filter are denied. This includes packet types such as IP and ARP. The result is that you have a Layer 2 filter but Layer 3 traffic is dropped. To avoid this, make sure you configure a filter to “permit any” traffic, thus changing the default action to permit for packets that are not denied by the other MAC filters.

Precedence Among Filters on the Same Layer

For most types of filters, a Foundry device applies filters based on the order in which you list them in a port’s inbound or outbound filter list. For example, if you apply three filters, 3, 2, and 1024 to port 1/1’s outbound filter list, the filters are applied in the following order: 3, 2, 1024.

You must configure the policies or filters before you can add them to a policy or filter group.

When you configure a policy or filter group, you must add all the policies or filters at the same time. You cannot edit policy or filter groups. To change a group, you must delete it, then add a new one.

NOTE: Foundry devices apply Layer 2 broadcast and multicast filters in ascending numerical order, beginning with 1.

Foundry Policies

On a Foundry device, a policy is a set of rules that defines how the device handles packets. The following table lists the types of policies you can configure on Foundry devices.

Table B.3: Foundry Policies

Policy Type	Supported on...			See page...
	Router	Switch	ServerIron	
Quality-of-Service (QoS) Policies	X	X	X	B-6
Layer 3 Policies				B-8
Protocol-based VLANs – either forward or drop Layer 3 traffic based on protocol (or, for IP subnet VLANs and IPX network VLANs, subnet or network address)	X	X	X	B-8
IP access policies – either forward or drop IP packets	X			B-9
Router acceleration policies – either switch (accelerate) IP or IPX packets or send them to a router		X[1]		B-18
Layer 4 Policies				B-39
TCP/UDP access policies – either forward or drop packets based on TCP or UDP port	X	X	X	B-20

Table B.3: Foundry Policies (Continued)

Policy Type	Supported on...			See page...
	Router	Switch	ServerIron	
Cache Server redirection policies – either redirect HTTP packets to cache servers or send the packets to the Internet **Note**: This type of policy applies only to the ServerIron Transparent Cache Switching (TCS) feature.			X	B-22

1. Router acceleration is supported only on the FastIron Backbone and TurboIron Backbone Layer 2 Switches.

Quality-of-Service Policies

Foundry devices support Quality-of-Service (QoS) through implementation of 802.1q prioritization. You can configure QoS policies for packets associated with the following items:

- Ports
- VLANs
- Static MAC entries
- Layer 4 sessions
- AppleTalk sockets.

The QoS levels provided by a Foundry device differ depending on the device:

- The Chassis devices (NetIron Internet Backbone router, BigIron, FastIron II, and chassis-based ServerIron) and the TurboIron/8 provide four weighted queues: 0 (normal) – 3 (highest priority).
- The Stackable devices (FastIron Workgroup, FastIron Backbone, TurboIron (4- to 6-port Gigabit Layer 2 Switch), NetIron, and stackable ServerIron) provide two QoS queues: normal and high.

NOTE: The ServerIron also provides a “cache” QoS queue. This queue applies only to Transparent Cache Switching (TCS) and enables the feature.

The default queue for all packets is normal (or 0). You can change QoS policy by placing a port, VLAN, static MAC entry, Layer 4 session, or AppleTalk socket into a higher queue. See “Configuring Basic Quality of Service” on page 2-1 for more information about the Foundry QoS algorithms.

Actions

QoS policies place packets in the specified queue for forwarding.

Scope

You can apply QoS policies to individual ports, VLANs, static MAC address, Layer 4 sessions, and AppleTalk sockets. If a port is a member of two or more of these items and has different priorities, the priorities are merged. However, the resulting priority is never lower than the highest priority.

Syntax

Use the following CLI commands or Web management interface panels to configure QoS policies.

Table B.4: QoS Policies

QoS Scope	CLI syntax	Web management links
Individual port	BigIron(config-if-1/1)# priority <0-7> TurboIron(config-if-1)# priority normal | high	Configure->Port
VLAN	BigIron(config-vlan-8)# priority <0-7> TurboIron(config-vlan-8)# priority normal | high	Configure->VLAN->Port
Static MAC address[1]	BigIron(config)# static-mac-address <mac-addr> ethernet <portnum> [priority <0-7>] [host-type | router-type] TurboIron(config)# static-mac-address <mac-addr> ethernet <portnum> [normal-priority | high-priority] [host-type | router-type]	Configure->Static Station
Layer 4 session	BigIron(config)# ip access-policy <num> priority <0-7> <ip-addr> <ip-mask> | any <ip-addr> <ip-mask> | any tcp | udp [<operator> [<tcp/udp-port-num>]] BigIron(config-if-1/1)# ip access-policy-group in | out <policy-list> FastIronII(config) ip policy <num> priority <0-7> tcp | udp <tcp/udp-port-num> global | local FastIronII(config-if-1/1) ip-policy <num> TurboIron(config)# ip access-policy <num> high | normal <ip-addr> <ip-mask> | any <ip-addr> <ip-mask> | any tcp | udp [<operator> [<tcp/udp-port-num>]] TurboIron(config-if-1)# ip access-policy-group in | out <policy-list> ServerIron(config)# ip policy <index> cache | fw | normal | high tcp | udp <tcp/udp-port-num> global | local ServerIron(config-if-1)# ip-policy <num> **Note**: You need this command only if you define local policies on the ServerIron.	Configure->IP->Access Policy Layer 4 QoS (link from the System configuration panel) System->Layer 4 QoS
AppleTalk socket	BigIron(config)# appletalk qos socket <number> priority <0-7>	Configure->IP->AppleTalk ->Socket QoS

1. You can configure static MAC addresses on Layer 2 Switches but not on Layer 3 Switches.

Layer 3 Policies

Layer 3 policies are rules that control transmission and receipt of packets based on Layer 3 routing protocol information in the packets. You can configure the following types of Layer 3 policies:

- Protocol-based VLANs
- IP access policies (same as IP filters)
- Router acceleration policies (IP and IPX switching policies)

Protocol-Based VLANs

Within an 802.1d port-based VLAN, you can configure protocol-based VLANs that define Layer 3 broadcast domains for specific protocols. By configuring a port as a member of a protocol VLAN, you establish a forwarding policy for that port.

For example, if you have a port-based VLAN that contains ports 1 – 12, you can configure some or all of the ports in the VLAN as an AppleTalk protocol VLAN. AppleTalk broadcast traffic received on one of the ports in the AppleTalk VLAN is broadcast to the other ports in the AppleTalk VLAN, but not to ports outside the AppleTalk VLAN.

When a port in protocol-based VLAN receives a packet, the device examines the Layer 3 information in the packet to determine whether the packet type is the same as the protocol type of the VLAN.

- If the packet is the same type as the protocol of the VLAN, the device forwards the packet.
- If the packet is another protocol type, the device drops the packet.

For example, when a port in an AppleTalk VLAN receives an AppleTalk packet, the port forwards the packet. The same port drops IPX packets, unless the port also is a member of an IPX VLAN.

IP subnet and IPX network VLANs are similar, except for these VLAN types the device examines the IP subnet or IPX network address.

- If the IP subnet or IPX network address matches the address of the IP subnet VLAN or IPX network VLAN, the device forwards the packet.
- If the subnet or network address does not match the VLAN, the device drops the packet.

See the “Configuring VLANs” chapter in the *Foundry Switch and Router Installation and Basic Configuration Guide* for VLAN configuration rules and examples.

Actions

A Foundry device forwards a packet if its Layer 3 protocol information matches the protocol VLAN’s protocol type, IP subnet, or IPX network; otherwise, the policy drops the packet.

Scope

The forwarding policy of a port-based VLAN applies only to that VLAN.

Syntax

Use the following CLI commands or Web management interface panels to configure VLAN policies.

Table B.5: VLAN Policies

Scope	CLI syntax	Web management links
VLAN type	BigIron(config)# vlan <vlan-id> by port BigIron(config-vlan-1)# [untagged] ethernet <portnum > [to | ethernet <portnum>]	Configure->VLAN->Port

NOTE: The **untagged** command applies only if you are removing 802.1q tagging from the ports in the VLAN. 802.1q tagging allows a port to be a member of multiple port-based VLANs. Ports in a port-based VLAN are tagged by default. The default tag is 8100 and is a global parameter.

IP Access Policies

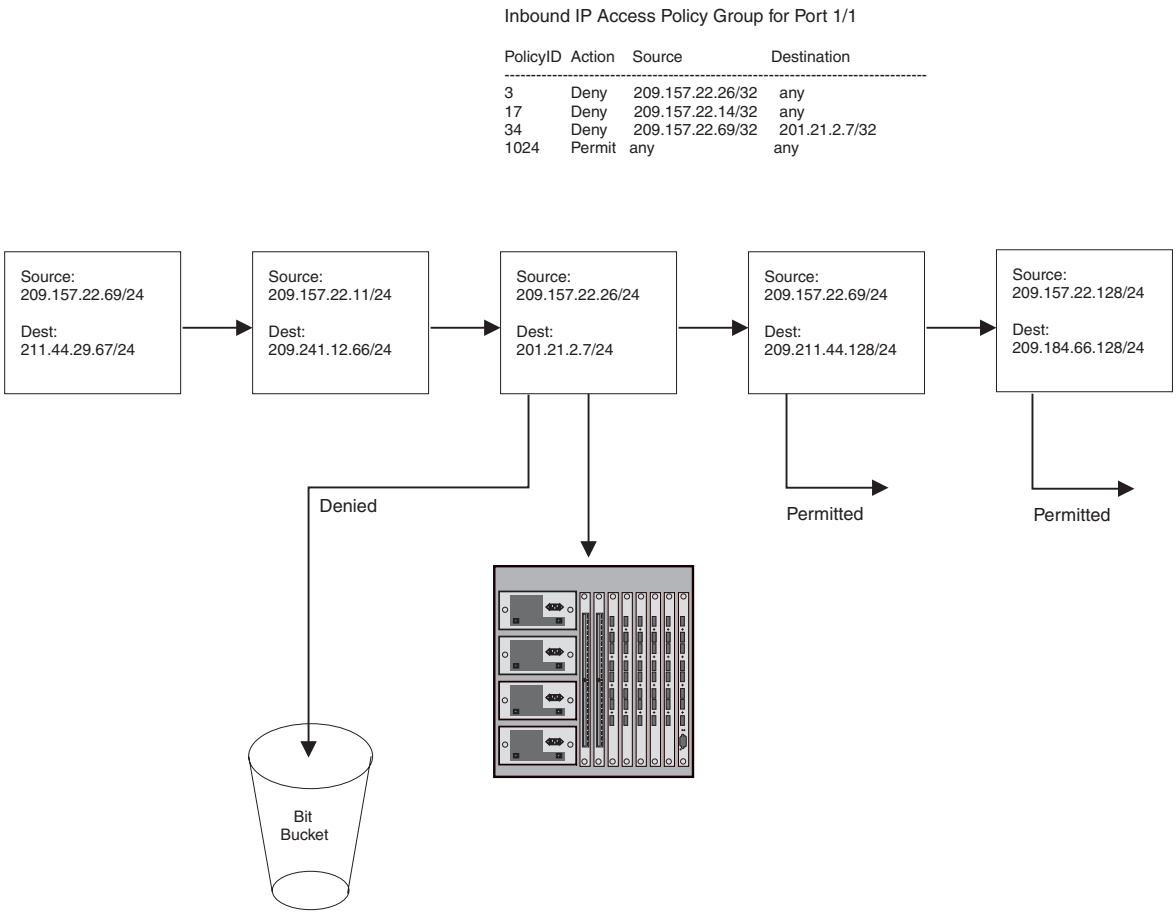
IP access policies are rules that determine whether the device forwards or drops IP packets. You create an IP access policy by defining an IP filter, then applying it to an interface. The filter consists of source and destination IP information and the action to take when a packet matches the values in the filter. You can configure an IP filter to permit (forward) or deny (drop) the packet.

You also can configure Layer 4 information in an IP filter. If you configure Layer 4 information, you are configuring a Layer 4 policy. See "TCP/UDP Access Policies" on page B-20.

You can apply an IP filter to inbound or outbound packets. When you apply the filter to an interface, you specify whether the filter applies to inbound packets or outbound packets. Thus, you can use the same filter on multiple interfaces and specify the filter direction independently on each interface.

Figure B.1 shows an example of an inbound IP access policy group applied to port 1 on slot 1 of a BigIron 8000 Layer 3 Switch. In this example, packets enter the port from left to right. The first three packets have entered the port and have been permitted or denied. The two packets on the left have not yet entered the port. When they do, they will be permitted. Since the last policy in the group is a "permit any" policy, all packets that do not match another policy are permitted. The "permit any" policy changes the default action to permit.

Figure B.1 IP access policies in inbound policy group for a port



Actions

IP access policies either forward or drop IP packets based on the IP source and IP destination addresses. You also can configure the policy to forward or drop a packet based on TCP/UDP port information. In this case, you are configuring a TCP/UDP access policy. See "TCP/UDP Access Policies" on page B-20.

Scope

You configure IP access policies globally, then apply them to individual ports. When you apply an IP policy to a port, you specify whether the policy applies to inbound or outbound packets. You can use the same policy in a port's inbound policy group and outbound policy group. When you configure a policy group, you must add all the policies to the group at one time. You cannot edit policy groups later. To change a policy group, you must delete the group and then add a new group.

Policies within the group are applied in positional order from left to right. Make sure you specify the filters in the order you want the device to apply them.

Syntax

Use the following CLI commands or Web management interface panels to configure IP access policies.

Table B.6: IP Access Policies

Foundry Product	CLI syntax	Web management links
NetIron Internet Backbone router, BigIron, FastIron II, TurboIron/8	BigIron(config)# ip access-policy <policy-num> permit | deny <ip-addr> <ip-mask> | any <ip-addr> <ipmask> | any tcp | udp [<operator> [<tcp/udp-port-num>]] [log] BigIron(config-if-1/1)# ip access-policy-group in | out <policy-list>	Configure->IP->Access Policy
NetIron	NetIron(config)# ip access-policy <policy-num> permit | deny <ip-addr> <ip-mask> | any <ip-addr> <ip-mask> | any tcp | udp [<operator> [<tcp/udp-port-num>]] [log] NetIron(config-if-1)# ip access-policy-group in | out <policy-list>	Configure->IP->Access Policy

Defining IP Access Policies

You can enhance network security by configuring IP access policies to explicitly permit or deny IP packets based on IP protocol, IP source and destination, IP protocol port, and even TCP or UDP application port.

NOTE: The device permits all IP packets by default. However, once you configure an IP access policy, the device denies all IP packets by default unless you explicitly permit them. Thus, if you want the device to permit all IP packets except the ones you filter out, you must configure the last IP access policy to permit all IP packets. If a packet does not match other filters (and thus is not denied), the packet matches the last filter and is permitted.

You can filter on the following IP protocols:

- ICMP
- IGMP
- IGRP
- OSPF
- TCP
- UDP

In addition, if you filter on TCP or UDP, you also can specify a particular application port (such as “HTTP” or “80”) or a logical expression consisting of an operator and port names or numbers. See the syntax descriptions below for details.

USING THE CLI

EXAMPLE:

To configure an IP access policy that globally accepts all FTP traffic without regard to network orientation, use the wildcard value ‘any' in place of an IP address and enter the following command:

```
BigIron(config)# ip access-policy 1 permit any any tcp eq ftp
```

EXAMPLE:

To configure an IP access policy that accepts only FTP traffic from a specific network, enter the following command:

```
BigIron(config)# ip access-policy 1 permit 192.38.5.54 255.255.255.0 195.38.5.53
255.255.255.0 tcp eq ftp
```

The following syntax applies to Chassis devices.

Syntax: ip access-policy <num> deny | permit <ip-addr> <ip-mask> | any <ip-addr> <ip-mask> | any icmp | igmp | igrp | ospf | tcp | udp | <num> [<operator> [<tcp/udp-port-num>]] [log]

Syntax: ip access-policy-group in | out <policy-list>

The following syntax applies to Stackable device.

Syntax: ip access-policy <num> deny | permit <ip-addr> <ip-mask> | any <ip-addr> <ip-mask> | any tcp | udp [<operator> [<tcp/udp-port-num>]] [log]

ip access-policy-group in | out <policy-list>

The <num> parameter is the policy number.

The **deny | permit** parameter specifies the action the router takes if a packet matches the policy.

- If you specify deny, the router drops the packet.
- If you specify permit, the router forwards the packet.

The <ip-addr> <ip-mask> **| any** <ip-addr> <ip-mask> **| any** parameters specify the source and destination IP addresses. If you specify a particular IP address, you also need to specify the mask for that address. If you specify **any** to apply the policy to all source or destination addresses, you do not need to specify **any** again for the mask. Make sure you specify a separate address and mask or **any** for the source and destination address.

The **icmp | igmp | igrp | ospf | tcp | udp |** <num> parameter specifies the IP protocol to which you are applying the policy. If you specify **tcp** or **udp**, you also can use the optional <operator> and <tcp/udp-port-num> parameters to fine-tune the policy to apply to specific TCP or UDP ports.

The <operator> parameter applies only if you use the **tcp** or **udp** parameter above. Use the <operator> parameter to specify the comparison condition for the specific TCP or UDP ports. For example, if you are configuring QoS 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**.
- **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.

- **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.

The **log** parameter applies only to deny policies. This parameter generates a Syslog entry for packets that are denied by the policy. See the "show logging" section of the "Show Commands" chapter in the *Foundry Switch and Router Command Line Interface Reference* for a description of how the timer for the entries works. Layer 2 MAC filters and IP access policies use the same timer, whereas Access Control Lists (ACLs) use a separate timer, but the timers work the same way. Thus, the description of how the ACL timer works also applies to the Layer 2 MAC filters and IP access policies.

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.

Figure B.2 and Figure B.3 on page B-14 show the CLI syntax for configuring an IP access policy.

Figure B.2 IP access policy syntax (1 of 2)

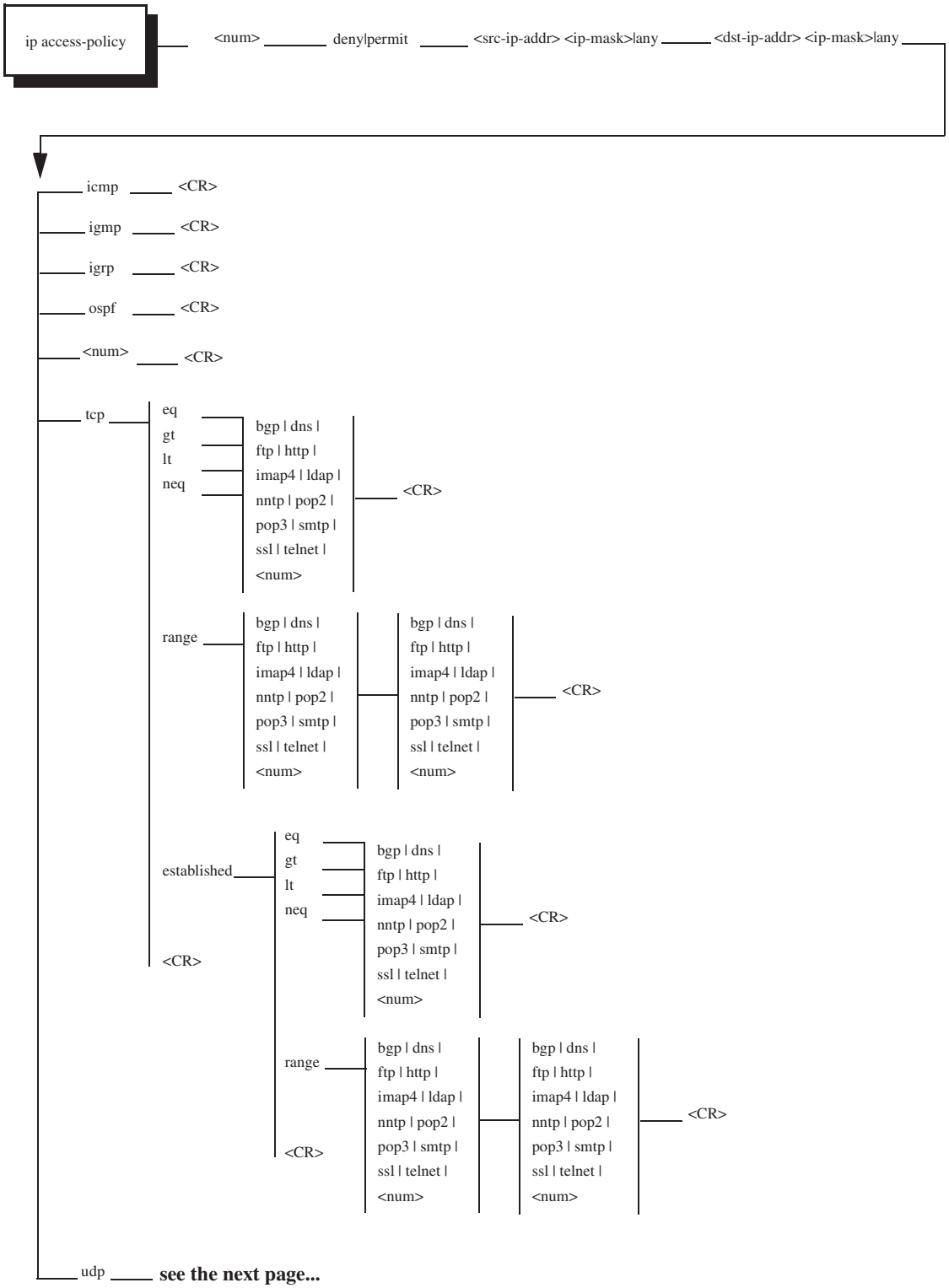
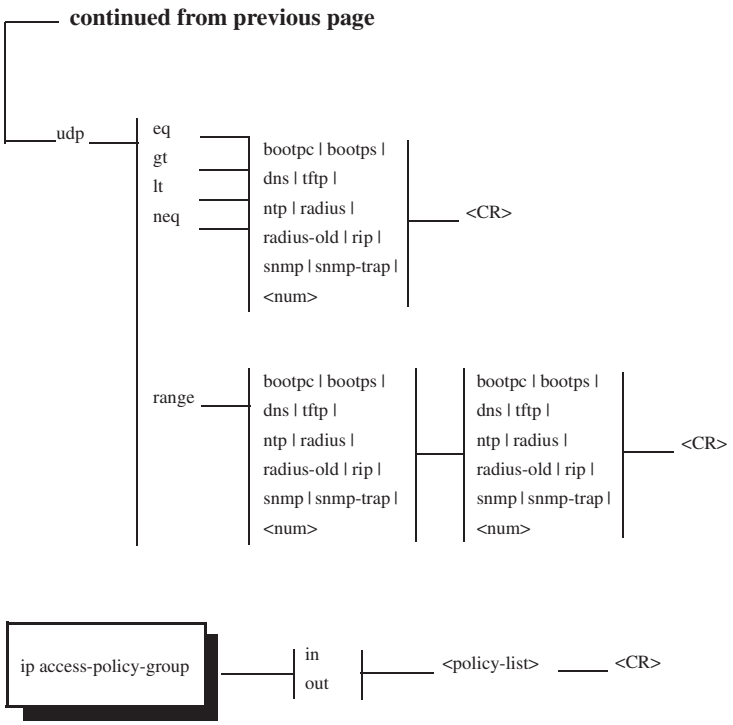


Figure B.3 IP access policy syntax (2 of 2)



USING THE WEB MANAGEMENT INTERFACE

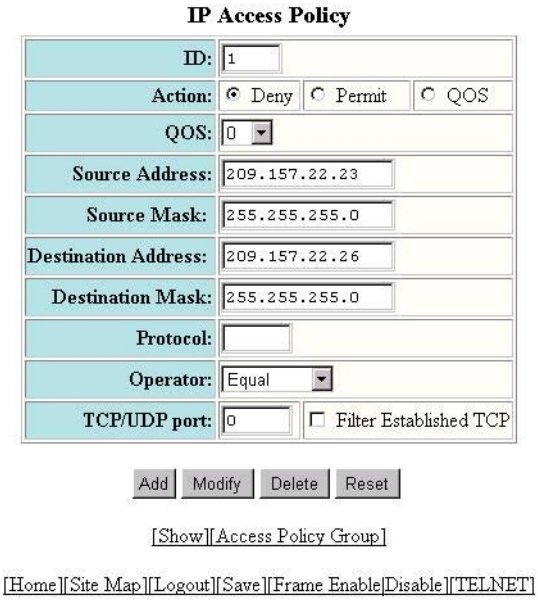
To add an IP access policy:

1. Log on to the device using a valid user name and password for read-write access. The System configuration panel is displayed.
2. Click on the plus sign next to Configure in the tree view to expand the list of configuration options.
   - On Layer 2 Switches – Click on the IP Access Policy link.
   - On Layer 3 Switches – Click on the plus sign next to IP in the tree view to expand the list of IP option links, then click the Access Policy link.

   If the device does not have any IP access policies, the IP Access Policy configuration panel is displayed, as shown in the following example.

   If an IP access policy is already configured and you are adding a new policy, click on the Add IP Access Policy link to display the IP Access Policy configuration panel, as shown in the following example.

   If you are modifying an existing IP access policy, click on the Modify button to the right of the row describing the policy to display the IP Access Policy configuration panel, as shown in the following example.



3. Enter an ID for the access policy.
4. Select Permit or Deny.

NOTE: If you select QoS, you are configuring a Layer 4 Quality of Service (QoS) policy. See “Assigning IP and Layer 4 Sessions to Priority Queues” on page 2-18.

---

5. Enter the source address and mask for the policy.

NOTE: You can specify the wildcard value “any” in the source and destination IP address and mask fields to allow all traffic. Entering 0.0.0.0 represents “any”. Likewise, to allow all protocols to be accepted by a filter, you can enter a single zero (0) in the protocol field.

---

6. Enter the destination address and mask for the policy.
7. If you want to filter on a specific IP protocol, select the protocol from the Protocol field’s pulldown menu. For example, to filter on TCP packets, select TCP. You can enter the protocol number or select one of the following:
   - ICMP
   - IGMP
   - IGRP
   - OSPF
   - TCP
   - UDP
8. If you selected TCP or UDP, you can select a comparison operator. Select the operator from the Operator field’s pulldown menu. You can select one of the following:
   - Greater – The policy applies to TCP or UDP port numbers greater than the port number or the numeric equivalent of the port name you specify.
   - Equal – The policy applies to the TCP or UDP port name or number you specify.

- Less – 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 specify.
- Not Equal – The policy applies to all TCP or UDP port numbers except the port number or port name you specify.
- Established (applies only to TCP) – 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.
- 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 specify. 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), specify the following: "23 53". The first port number in the range must be lower than the last number in the range.

9. If you selected a comparison operator, enter the port number in the TCP/UDP port field. For example, if you selected TCP and Equal and you want to filter on HTTP traffic, enter the value 80 (the well-known port number for HTTP).

NOTE: You must enter the port's number instead of the well-known name.

10. Click the Add button to save the change to the device's running-config file.
11. 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.
12. Go to "Applying IP Access Policies to Ports" on page B-16. The policy does not take effect until you apply it to a port.

Modifying or Deleting an IP Access Policy

To modify or delete an IP access policy:

1. Log on to the device using a valid user name and password for read-write access. The System configuration dialog is displayed.
2. Click on the plus sign next to Configure in the tree view to expand the list of configuration options.
   - On Layer 2 Switches – Click on the IP Access Policy link.
   - On Layer 3 Switches – Click on the plus sign next to IP in the tree view to expand the list of IP option links, then click the Access Policy link.
3. When the IP Access Policy table appears, click the Modify or Delete button on the row for the policy you want to modify or delete.
4. 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.
5. If you modified a policy that is not yet assigned to a port, go to "Applying IP Access Policies to Ports" on page B-16. The policy does not take effect until you apply it to a port.

Applying IP Access Policies to Ports

Once you define an IP access policy, you can apply it to the inbound or outbound traffic on a port by configuring an IP access policy group for the port. Policies within the group are applied in positional order from left to right. Make sure you specify the policies in the order you want the device to apply them.

USING THE CLI

To assign IP access policies 2, 3, and 5 to port 1 on module 2 of a Chassis device, enter the following commands:

```
BigIron(config)# interface e 2/1
BigIron(config-if-2/1)# ip access-policy-group in 2 3 5
```

Syntax: ip access-policy-group in | out <policy-list>

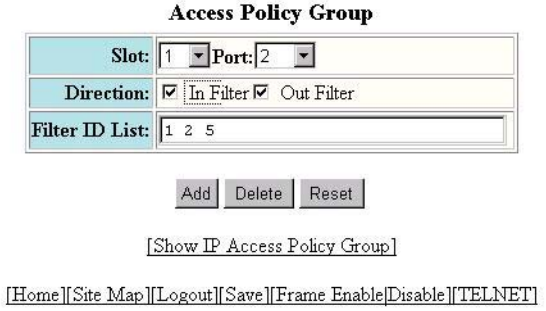
You also can specify policy ranges. For example, to apply policies 1 – 3, policy 9, and policies 11 – 25 to port 2/4's outbound policy group, enter the following command:

```
BigIron(config)# interface ethernet 2/4
BigIron(config-if-2/4)# ip access-policy-group out 1 to 3 9 11 to 25
```

USING THE WEB MANAGEMENT INTERFACE

To assign IP filters 1, 2, and 5 to port 2 on module 1 of a Chassis device:

1. Log on to the device using a valid user name and password for read-write access. The System configuration panel is displayed.
2. Click on the plus sign next to Configure in the tree view to expand the list of configuration options.
3. Click on the plus sign next to IP in the tree view to expand the list of IP option links.
4. Click the Access Policy link.
5. Click the Access Policy Group link.
   - If the device does not have any IP access policy groups, the Access Policy Group configuration panel is displayed, as shown in the following example.
   - If an IP access policy group is already configured and you are adding a new policy group, click on the Add IP Access Policy Group link to display the Access Policy Group configuration panel, as shown in the following example.
   - If you are modifying an existing IP access policy group, click on the Modify button to the right of the row describing the policy group to display the IP Access Policy Group configuration panel, as shown in the following example.



6. Select the port (and slot if applicable) to which you are assigning the access policies.
7. Select either or both the In Filter and Out Filter options.
   - Selecting In Filter applies the access policies to all incoming traffic on the port.
   - Selecting Out Filter applies the access policies to all outgoing traffic on the port.
   - Selecting both options applies the access policies to both incoming and outgoing traffic.
8. Enter the access policy IDs in the Filter ID List field. To enter a range, enter the first policy number in the range, a space, a dash, another space, and then the second policy number. For example, enter "1 – 4" to specify the range 1 – 4.

NOTE: When specifying a range, you must use spaces on either side of the dash.

9. Click the Add button to save the change to the device's running-config file.
10. 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.

Router Acceleration Policies

You can configure Foundry Layer 2 Switches and Layer 3 Switches to perform router acceleration by enabling Layer 3 IP or IPX 3 switching. When you enable router acceleration, the device switches all IP or IPX route packets by default after learning the outbound ports for the packets. By switching an IP or IPX packet, the Foundry Layer 2 or Layer 3 Switch provides increased network performance by bypassing slow routers.

NOTE: Router acceleration is supported only on the FastIron Backbone and TurboIron Backbone Layer 2 Switches.

NOTE: Foundry Layer 3 Switches route at wire speed. Router acceleration is useful for routers that do not route at wire speed.

Switching the IP or IPX packets to bypass the router is the default router acceleration policy. You can modify the policy by adding IP or IPX filters. The CLI commands and Web management panels for router acceleration filters are the same as those used for IP and IPX forward filters. You configure the filters globally, then apply them to specific ports. When you apply an IP or IPX filter to a port, you specify whether the filter applies to inbound packets or outbound packets.

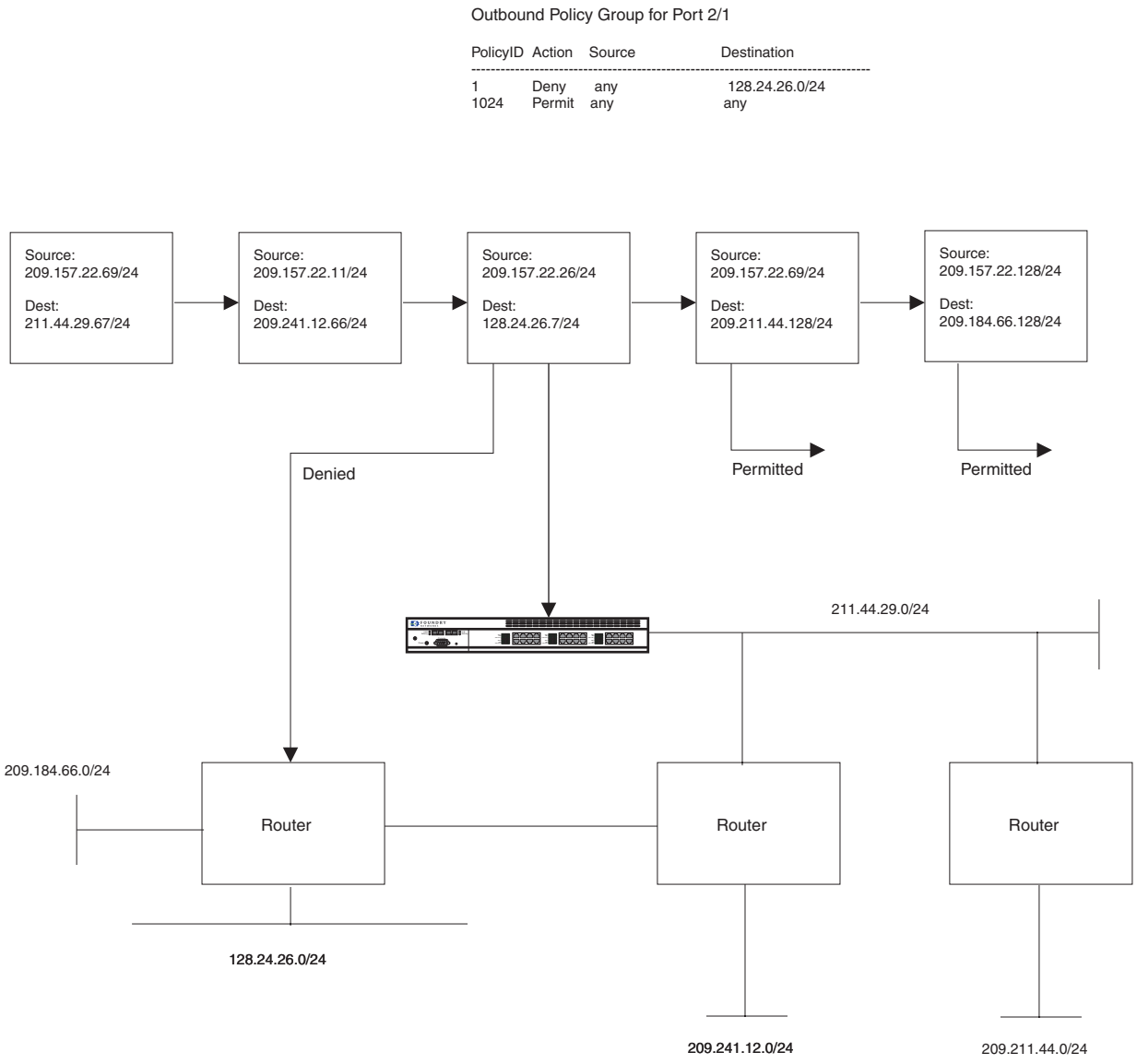
NOTE: A Foundry device can either route or switch IP or IPX, but cannot be configured to both route and switch the same protocol.

IP and IPX forwarding filter behavior differs depending on whether the device is switching or routing:

- If the device is routing, the behavior is as described in “IP Access Policies” on page B-9.
- If the device is switching, the device accelerates packets that match a filter with a permit action. The device does not drop packets that match a filter with a deny action; instead, the device sends these packets to the router.

Figure B.4 shows an example of router acceleration policies. In this example, all outbound IP traffic on Port 2/1 except traffic destined for IP subnet 128.24.26.0/24 is permitted to be directly switched at Layer 3, bypassing the router. The traffic that is denied Layer 3 switching by policy 1 is sent to the router instead. The policies in this example are applied to a port’s outbound policy group. Therefore, all packets enter the Foundry device before being filtered.

Figure B.4 Router acceleration policies



Actions

Router acceleration policies either accelerate (permit) packets or send them to the router (deny acceleration). they do not drop packets.

Scope

You define router acceleration policies globally, then apply them to individual ports.

Syntax

Use the following CLI commands or Web management interface panels to configure router acceleration policies.

Table B.7: Router Acceleration Policies

Foundry Product	CLI syntax	Web management links
FastIron Backbone, TurboIron Backbone	TurboIron(config)# ip policy <num> priority high | normal tcp | udp <tcp/udp-port-num> global | local TurboIron(config-if-1)# ip policy-group in | out <policy-list> TurboIron(config)# ipx forward-filter <filter-num> permit | deny <source-network-number> | any <source-node-number> | any <destination-network-number> | any TurboIron(config-if-1)# ipx forward-filter-group in | out <filter-list> <destination-node-number> | any <destination-socket-number> | any	Configure->IP->Access Policy Configure->IPX->Forward Filter

NOTE: You must enable router acceleration before the feature or the policies will take effect. To enable IP router acceleration, enter the **ip-route-accelerating** or **ipx-route-accelerating** command at the global CONFIG level of the CLI.

Layer 4 Policies

Layer 4 policies are rules that control transmission and receipt of packets based on Layer 4 transport information. You can configure the following types of Layer 4 policies:

- TCP/UDP access policies (same as TCP/UDP filters)
- Cache server redirection policies (used by the ServerIron's Transparent Cache Switching feature)

TCP/UDP Access Policies

TCP/UDP access policies are IP filters that contain Layer 4 information. Layer 4 policies enable you to forward or drop packets for individual Layer 4 applications, giving you finer access control. You do not need to completely block an IP address to deny certain types of traffic from that address. You can selectively allow some types of traffic while dropping others. For example, you can configure a Layer 4 policy to drop web (HTTP) packets from a host but allow all other traffic from the host.

You can filter on the following Layer 4 application types:

- ICMP
- IGMP
- IGRP
- OSPF
- TCP
- UDP

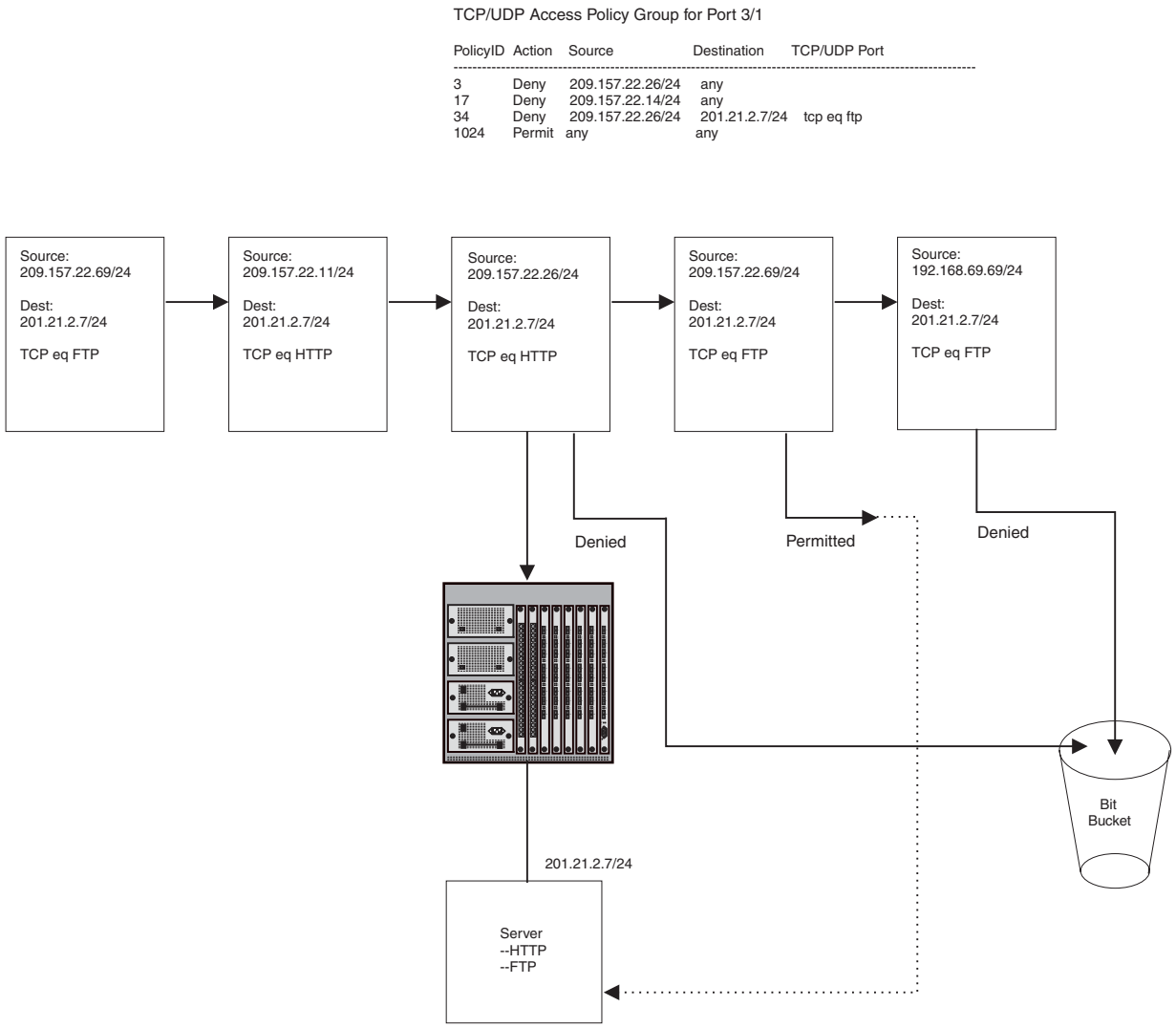
For TCP and UDP, you also specify an operator and the port number or well-known name for the port. For example, if you want to filter on FTP traffic, you configure the filter to match on packets that contain the TCP application port number for FTP.

You can configure Layer 4 policies on all Foundry devices.

- If you configure them on Layer 2 Switches and Layer 3 Switches, you specify the source and destination IP address of the hosts or servers for which you are controlling access.
- If you configure Layer 4 policies on a ServerIron configured for Server Load Balancing (SLB), you specify the virtual IP address (VIP) associated with the real servers.

Figure B.5 shows an example of TCP/UDP access policies. Although this example does not explicitly identify these policies as inbound policies or outbound policies, when you apply the policies to individual ports you specify whether they are for inbound or outbound traffic.

Figure B.5 TCP/UDP Access Policies



Actions

TCP/UDP access policies forward (permit) or drop (deny) IP packets based on the Layer 4 application information in the packets.

Scope

You configure TCP/UDP access policies globally, then apply them to individual ports. When you apply a TCP/UDP policy to a port, you specify whether the policy applies to inbound or outbound packets. You can use the same policy in a port's inbound policy group and outbound policy group. When you configure a policy group, you must add all the policies to the group at one time. You cannot edit policy groups later. To change a policy group, you must delete the group and then add a new group.

Policies within the group are applied in positional order from left to right. Make sure you specify the filters in the order you want the device to apply them.

Syntax

Use the following CLI commands or Web management interface panels to configure TCP/UDP access policies.

Table B.8: TCP/UDP Access Policies

Foundry Product	CLI syntax	Web management links
NetIron Internet Backbone router, BigIron, FastIron II, TurboIron/8	BigIron(config)# ip access-policy <policy-num> permit | deny <ip-addr> <ip-mask> | any <ip-addr> <ip-mask> | any tcp | udp [<operator> [<tcp/udp-port-num>]] [log] BigIron(config-if-1/1)# ip access-policy-group in | out <policy-list> FastIronII(config) ip policy <policy-num> permit | deny tcp | udp <tcp/udp-port-num> global | local FastIronII(config-if-1/1) ip-policy <num>	Configure->IP->Access Policy
FastIron Workgroup, FastIron Backbone, TurboIron, NetIron	NetIron(config)# ip access-policy <policy-num> permit | deny <ip-addr> <ip-mask> | any <ip-addr> <ip-mask> | any tcp | udp [<operator> [<tcp/udp-port-num>]] [log] NetIron(config-if-1)# ip access-policy-group in | out <policy-list> FastIron(config)# ip policy <num> priority high | normal tcp | udp <tcp/udp-port-num> global | local FastIron(config-if-1)# ip-policy <num>	Configure->IP->Access Policy
ServerIron	ServerIron(config)# ip filter <filter-num> permit | deny <IP-addr> <ip-mask> | any <IP-addr> <ip-mask> | any [icmp | tcp | udp | <num>] [<operator>] [<tcp/udp-port-num>]	Configure->IP->Access Policy

Cache Server Redirection Policies

Cache server redirection polices apply only to a ServerIron configured for Transparent Cache Switching (TCS). You configure the redirection policies to enable TCS.

Actions

A cache server redirection policy either redirects HTTP traffic to a cache server (permits) or sends the traffic to the Internet (denies).

Scope

You can enable TCS globally or you can enable it locally, on specific ports. If you enable TCS locally, make sure you enable TCS on the ports that are connected to the Internet. Enable TCS for outbound traffic.

Syntax

Use the following CLI commands or Web management interface panels to configure cache server redirection policies.

Table B.9: Cache Server Redirection Policies

Foundry Product	CLI syntax	Web management links
ServerIron	ServerIron(config)# ip policy <policy-num> <cache> tcp | udp <tcp/udp-port-num> global | local ServerIron(config-if-18)# ip-policy <policy-num>	Layer 4 QoS (link from the System configuration panel)

Foundry Filters

A filter is a set of comparison values and an action. If a packet matches the set of values in the filter, the Foundry device takes the action specified in the filter. Foundry devices provide filters for Layer 2, Layer 3, and Layer 4. A filter looks at the appropriate fields in a packet to compare information related to one of the layers. For example, MAC filters look at the source and destination MAC address and, optionally, at the encapsulation information. IPX filters look at the source and destination network and socket information but do not look at the MAC information.

The following table lists the various types of filters you can configure on Foundry devices.

Table B.10: Foundry Filters

Filter Type	Supported on...					See page...
	FES	FESX	Router	Switch	ServerIron	
Layer 2 Filters						B-24
MAC filters	X	X	X	X	X	B-24
Broadcast filters			X	X	X	B-25
Multicast filters			X	X	X	B-26
Address-lock filters	X	X	X	X	X	B-26
Layer 3 Filters						B-27
IP switching filters and IPX switching filters (same as router acceleration policies)	X	X (IP only)	X	X		B-18
IP forwarding filters (same as IP access policies)	X	X	X			B-9
RIP route filters	X	X	X			B-28
RIP neighbor filters	X	X	X			B-29
IPX forwarding filters	X		X			B-30
IPX RIP filters	X		X			B-31
IPX SAP filters	X		X			B-31

Table B.10: Foundry Filters (Continued)

Filter Type	Supported on...					See page.. .
	FES	FESX	Router	Switch	ServerIron	
AppleTalk zone filters	X		X			B-33
AppleTalk network filters	X		X			B-33
BGP address filters			X			B-34
BGP AS-path filters			X			B-35
BGP community filters			X			B-36
RIP redistribution filters	X	X	X			B-37
OSPF redistribution filters	X	X	X			B-38
BGP redistribution filters			X			B-38
Layer 4 Filters						B-39
TCP/UDP forwarding filters (same as TCP/UDP access policies)	X	X	X	X	X	B-20
Cache server redirection filters (same as cache server redirection policies)					X	B-22

Layer 2 Filters

Layer 2 filters control a Foundry device's receipt of packets based on MAC address information. Foundry devices provide the following types of Layer 2 filters:

- MAC address filters
- Address-lock filters

MAC Filters

MAC filters forward or drop incoming packets based on the following information:

- Source MAC address
- Destination MAC address
- Encapsulation type and EtherType (optional)

A packet whose Layer 2 information matches the filter is either permitted (forwarded) or denied (dropped). You define a MAC filter on the global level, then apply it to an interface. The filter applies only to incoming traffic on the interface.

NOTE: MAC filters do not block management access to the Foundry device. For example, if you apply a filter to block a specific host, the filter blocks switch traffic from the host but does not prevent the host from establishing a management connection to the device through Telnet. To block management access, use an Access Control List (ACL). See "Access Control List" on page 6-1.

Action

MAC filters forward (permit) or drop (deny) packets.

Scope

You configure MAC filters globally, then apply them to individual ports. The filters do not take effect until applied to specific ports. MAC filters apply only to incoming packets.

Syntax

Use the following CLI commands or Web management interface panels to configure MAC filters.

Table B.11: MAC Filters

CLI syntax	Web management links
BigIron(config)# mac filter <filter-num> permit | deny any | <H.H.H> any | <H.H.H> etype | llc | snap <operator> <frame-type> BigIron(config-if-1/1)# mac-filter-group <filter-list>	Configure->MAC Filter

Broadcast Filters

Broadcast filters are outbound filters that drop Layer 2 broadcast packets that match the filter criteria. You can filter on all broadcast traffic or on IP UDP broadcast traffic only. You also can specify a VLAN ID so that broadcasts are dropped only for the specified VLAN.

You can configure up to eight broadcast filters.

NOTE: Broadcast filters are applied in numerical order, beginning with filter 1.

Action

Broadcast filters forward (permit) or drop (deny) packets.

Scope

You configure broadcast filters globally, then apply them to individual ports. The filters do not take effect until applied to specific ports. The filters apply only to outbound traffic. The **exclude-ports** command specifies the ports on which you are excluding the filtered multicast packets.

Syntax

Use the following CLI commands or Web management interface panels to configure broadcast filters.

Table B.12: Broadcast Filters

CLI syntax	Web management links
BigIron(config)# broadcast filter <filter-id> any | ip udp [vlan <vlan-id>] exclude-ports ethernet <portnum> to <portnum> Or exclude-ports ethernet <portnum> ethernet <portnum>	Not available

NOTE: This is the same command syntax as that used for configuring port-based VLANs. Use the first command for adding a range of ports. Use the second command for adding separate ports (not in a range). You also can combine the syntax. For example, you can enter **exclude-ports ethernet 1/4 ethernet 2/6 to 2/9**.

Multicast Filters

Multicast filters are outbound filters that apply to packets that have a Layer 2 multicast address in the destination MAC address field. You can configure multicast filters to filter on all multicast addresses or a specific multicast address.

You can configure up to eight multicast filters.

NOTE: Multicast filters are applied in numerical order, beginning with filter 1.

Action

Multicast filters forward (permit) or drop (deny) packets.

Scope

You configure multicast filters globally, then apply them to individual ports. The filters do not take effect until applied to specific ports. The filters apply only to outbound traffic. The **exclude-ports** command specifies the ports on which you are excluding the filtered multicast packets.

Syntax

Use the following CLI commands or Web management interface panels to configure multicast filters.

Table B.13: Multicast Filters

CLI syntax	Web management links
BigIron(config)# multicast filter <filter-id> any | ip udp mac <multicast-address> | any [mask <mask>] [vlan <vlan-id>] exclude-ports ethernet <portnum> to <portnum> Or exclude-ports ethernet <portnum> ethernet <portnum>	Not available

NOTE: This is the same command syntax as that used for configuring port-based VLANs. Use the first command for adding a range of ports. Use the second command for adding separate ports (not in a range). You also can combine the syntax. For example, you can enter **exclude-ports ethernet 1/4 ethernet 2/6 to 2/9**.

Address-Lock Filters

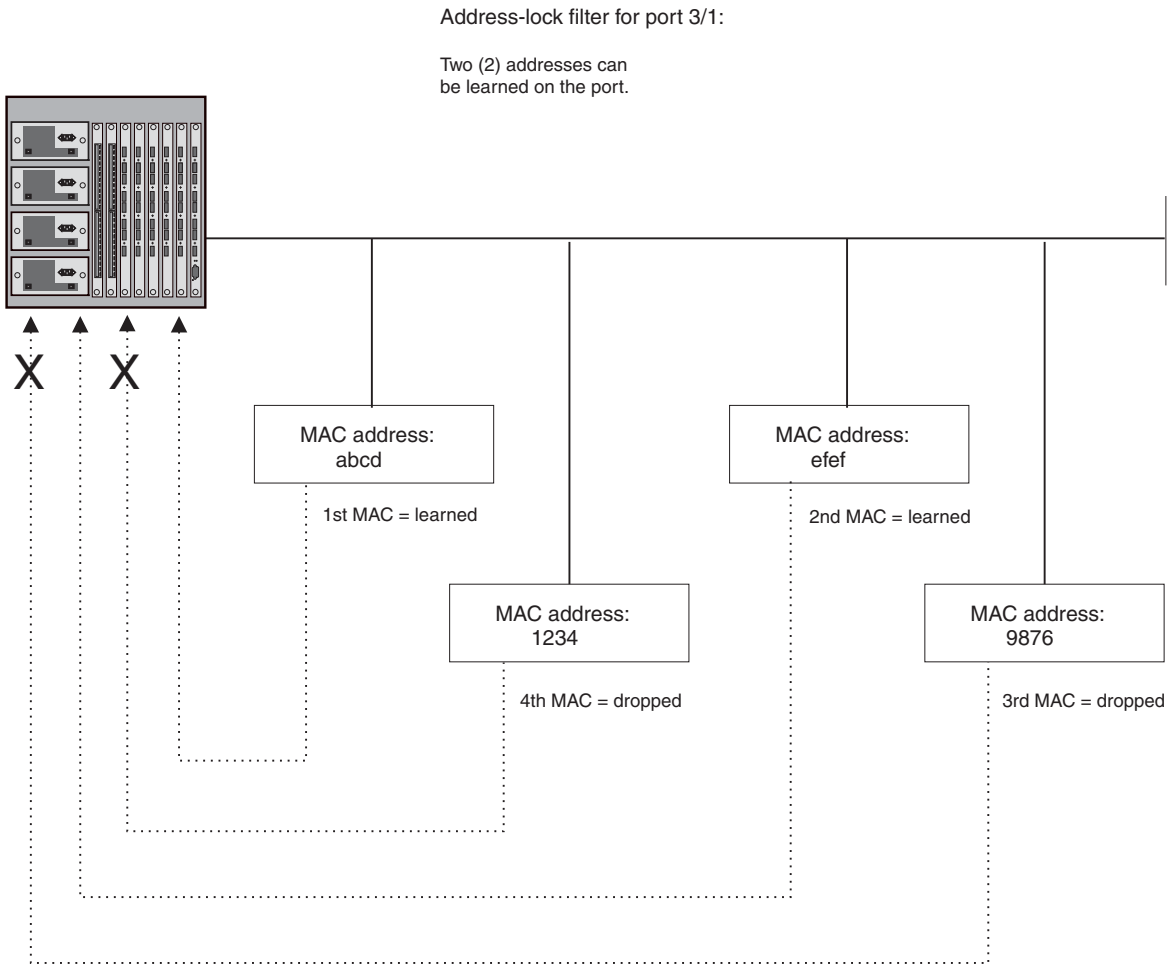
Address-lock filters limit the number of MAC addresses that can be learned on a port. The port forwards only those packets that contain one of the source MAC addresses learned by the port. The port drops other packets. In addition, the device generates an SNMP trap for other packets received by the port.

Figure B.6 shows an example of an address-lock filter. In this example, the Foundry device is configured to learn only two MAC addresses on port 3/1. After the device learns two addresses, port 3/1 can forward only a packet whose source address is one of the two learned addresses. The port drops all other packets. This applies even to MAC broadcasts. If one of the packets learned on the port is not addressed to the MAC broadcast address, the port cannot forward MAC broadcasts.

The device learns MAC addresses from the source-MAC-address field of inbound packets received on the port.

NOTE: The FastIron Edge Switch does not support address-lock filters on static trunk ports or ports on which link-aggregation is enabled.

Figure B.6 Address-lock filter



Actions

Forward (permit) only those packets with a MAC address that the port has learned. Deny all other packets.

Scope

You configure a lock address filter globally, but you also specify the port as part of the filter.

Syntax

Use the following CLI commands or Web management interface panels to configure address-lock filters.

Table B.14: Address-Lock Filters

CLI syntax	Web management links
BigIron(config)# lock-address ethernet <portnum> addr-count <num>	Configure->Port

Layer 3 Filters

Layer 3 filters control a Foundry device's transmission and receipt of packets based on routing protocol information in the packets. Foundry devices provide the following types of Layer 3 filters:

- IP forwarding filters (same as IP access policies, see "IP Access Policies" on page B-9)

- RIP route filters
- RIP neighbor filters
- IPX forwarding filters
- IPX RIP route and neighbor filters
- IPX SAP service filters
- AppleTalk zone filters
- AppleTalk network filters
- BGP route address filters
- BGP route AS-path filters
- BGP route community filters
- RIP redistribution filters
- OSPF redistribution filters
- BGP redistribution filters
- Router accelerator (IP and IPX switching) filters (same as router acceleration policies, see “Router Acceleration Policies” on page B-18)

IP Filters

IP filters control the IP packets that the Foundry device sends and receives and the routes that the device learns or advertises. IP forwarding filters (IP Access policies) control transmission and receipt of IP packets, while RIP route and neighbor filters control the routes that the device leans or advertises. Route filters filter on specific network addresses while neighbor filters filter on the IP addresses of the RIP neighbors.

IP Forwarding Filters

IP forwarding filters determine whether to forward or drop an IP packet. IP forwarding filters on a Foundry Layer 2 Switch or Layer 3 Switch are called “IP access policies”. See “IP Access Policies” on page B-9.

RIP Route Filters

RIP route filters control the routes that a Foundry device learns and advertises. Figure B.7 shows an example of a port with RIP route filters. The port has filters for the inbound direction and the outbound direction. Notice that the same filter can be used for both directions. The inbound filters control the routes that the device learns; denied routes are not learned by the device. Outbound filters control the routes that the device advertises; denied routes are not advertised to RIP neighbors.

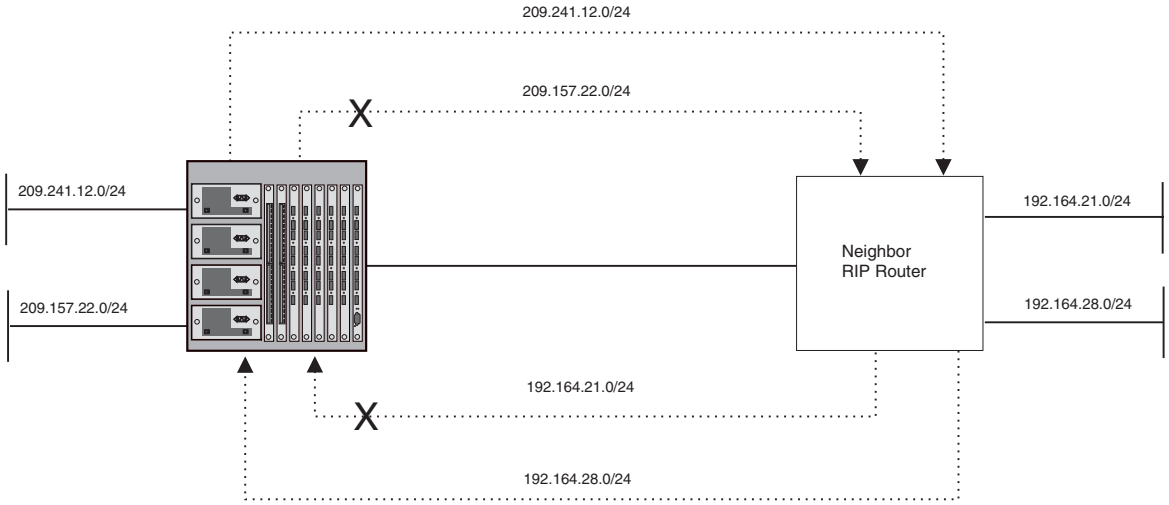
Figure B.7 RIP route filters

Outbound IP/RIP Route Filter Group for Port 4/1

FilterID	Action	Source
1	Deny	209.157.22.0/24
1024	Permit	any

Inbound IP/RIP Route Filter Group for Port 4/1

FilterID	Action	Source
2	Deny	192.164.21.0/24
1024	Permit	any



Actions

- An RIP route filter applied to outbound traffic on a port permits or denies advertisement of routes.
- An RIP route filter applied to inbound traffic on a port permits or denies learning of the route. When the device learns an RIP route, the route is added to the RIP route table.

Scope

You configure RIP route filters globally, then apply them to specific ports.

Syntax

Use the following CLI commands or Web management interface panels to configure RIP route filters.

Table B.15: RIP Route Filters

CLI syntax	Web management links
BigIron(config-rip-router)# filter <filter-num> permit | deny <source-ip-address> | any <source-mask> | any BigIron(config-if-1/1)# ip rip filter-group in | out <filter-list>	Configure->RIP->Route Filter

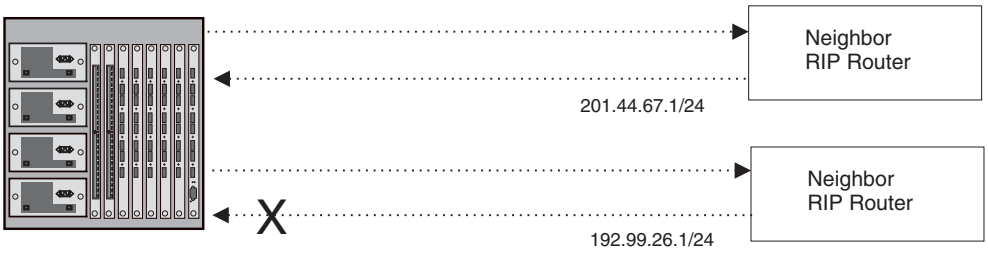
RIP Neighbor Filters

RIP neighbor filters specify the RIP neighbors the Foundry device can receive updates from or send updates to. You identify the neighbor by specifying its IP address in the filter. Figure B.8 shows an example of an RIP neighbor filter. In this example, the Foundry device is configured to drop all RIP advertisements from the RIP neighbor 192.99.26.1/24. Since this is an outbound filter, the filter does not affect advertisements received by the Foundry device from 192.99.26.1/24. The Foundry device can still learn RIP routes from this neighbor.

Figure B.8 RIP neighbor filters

```
Inbound IP/RIP Neighbor Filter for Port 4/3

FilterID  Action    Source
-------------------------------------------------------------
1         Deny      192.99.26.1/24
1024      Permit    any
```



Actions

- An RIP neighbor filter applied to outbound traffic on a port permits or denies advertisement of routes.
- An RIP neighbor filter applied to inbound traffic on a port permits or denies learning of the routes advertised by the neighbor. When the device learns an RIP route, the route is added to the RIP route table.

Scope

You configure RIP neighbor filters globally. They are automatically applied to all RIP ports as soon as you configure them.

Syntax

Use the following CLI commands or Web management interface panels to configure RIP neighbor filters.

Table B.16: RIP Neighbor Filters

CLI syntax	Web management links
BigIron(config-rip-router)# neighbor <filter-num> permit | deny <source-IP-address> | any	Configure->RIP->Neighbor Filter

IPX Filters

IPX filters control transmission and receipt of IPX packets, IPX RIP routes, and IPX Service Advertisement Protocol (SAP) messages. IPX forwarding filters filter on source and destination IPX address and socket information. IPX RIP filters filter based on a route's network address. IPX SAP filters filter based on server type and server name.

IPX Forwarding Filters

IPX forwarding filters control forwarding of IPX packets.

Action

- An IPX forward filter applied to inbound packets forwards or drops IPX packets received on the port.
- An IPX forward filter applied to outbound traffic forward or drops IPX packets sent to the port for forwarding.

Scope

You configure IPX forwarding filters globally, then apply them to specific ports.

Syntax

Use the following CLI commands or Web management interface panels to configure IPX forwarding filters.

Table B.17: IPX Forwarding Filters

CLI syntax	Web management links
BigIron(config)# ipx forward-filter <filter-num> permit | deny <source-network-number> | any <source-node-number> | any <destination-network-number> | any <destination-node-number> | any <destination-socket-number> | any BigIron(config-if-1/1)# ipx forward-filter-group in | out <filter-list>	Configure->IPX->Forward Filter

IPX RIP Filters

IPX RIP filters control the IPX routes that the Foundry device learns or advertises.

Actions

- An IPX RIP filter applied to inbound packets learns or drops IPX routes received on the port.
- An IPX RIP filter applied to outbound packets advertises or does not advertise IPX routes.

Scope

You configure IPX RIP filters globally, then apply them to specific ports.

Syntax

Use the following CLI commands or Web management interface panels to configure IPX RIP filters.

Table B.18: IPX RIP Filters

CLI syntax	Web management links
BigIron(config)# ipx rip-filter <filter-num> permit | deny <network-number> | any <network-mask> | any BigIron(config-if-1/1)# ipx rip-filter-group in | out <filter-list>	Configure->IPX->RIP Filter

IPX SAP Filters

IPX Service Advertisement Protocol (SAP) filters control client access to IPX servers.

Actions

- An IPX SAP filter applied to inbound packets learns or drops advertisements for the specific services.
- An IPX SAP filter applied to outbound traffic advertises or does not advertise services.

Scope

You configure IPX SAP filters globally, then apply them to specific ports.

Syntax

Use the following CLI commands or Web management interface panels to configure IPX SAP filters.

Table B.19: IPX SAP Filters

CLI syntax	Web management links
BigIron(config)# ipx sap-filter <filter-num> permit | deny <server-type> | any <server-name> | any BigIron(config-if-1/1)# ipx sap-filter-group in | out <filter-list>	Configure->IPX->SAP Filter

Appletalk Filters

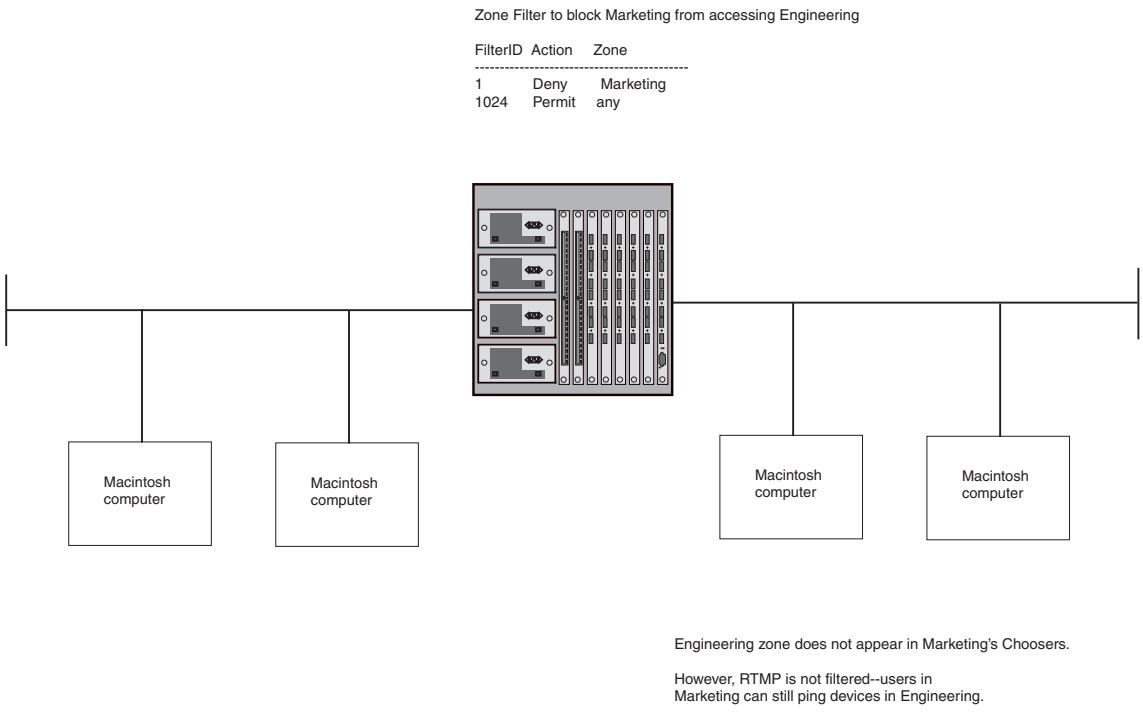
AppleTalk filters control access to AppleTalk zones and networks.

- AppleTalk zone filters permit or deny advertisement of zone names but allow network information to be learned and forwarded. Users cannot see the zone names in their Choosers but you can ping the networks. Zone filters are quite useful for reducing protocol overhead caused by “chatty” AppleTalk traffic. Use zone filtering to block information from a specific router to Macintosh computers.
- AppleTalk network filters also can filter network information. When you configure an AppleTalk zone filter to deny zones, you can configure the filter to also deny the network information. To configure an AppleTalk filter to filter network information, use the RTMP filtering option with the filter.

Figure B.9 shows an example of an AppleTalk zone filter. In this example, Macintosh computers in the Marketing zone cannot see the Engineering zone. RTMP filtering is not used on this filter. Therefore, users in the Marketing zone can still ping individual devices in the Engineering zone. However, the overhead caused by unnecessary zone information exchanges between the two groups is eliminated.

To prevent users in the Marketing zone from even pinging individual devices in the Engineering zone, the RTMP filtering option can be used with the filter.

Figure B.9 AppleTalk zone filter



Appletalk Zone Filters

AppleTalk zone filters let you secure access to an AppleTalk zone. The filter controls whether the Foundry Layer 3 Switch includes the zone in replies to a MAC chooser's ZIP GetZoneList request.

Actions

An AppleTalk zone filter permits (advertises) or denies (does not advertise) the specified zone. The zone does not appear in MAC user's choosers but you can still ping the networks that belong to the zone.

NOTE: Unlike other filters, the default action for AppleTalk filters does not change from permit to deny when you create a filter. To permit only specific zones and deny all others, create permit filters for the zones you want to permit, then use the following command to create a deny filter for all other zones: **appletalk deny zone additional-zones**.

Scope

You configure and apply AppleTalk zone filters on individual ports.

Syntax

Use the following CLI commands or Web management interface panels to configure AppleTalk zone filters.

Table B.20: AppleTalk Zone Filters

CLI syntax	Web management links
BigIron(config-if-1/1)# appletalk permit zone <string>	Configure->AppleTalk->Zone Filter
BigIron(config-if-1/1)# appletalk deny zone <string> | additional-zones rtmp-filtering | no-rtmp-filtering	Configure->AppleTalk->Additional Zone Filter

NOTE: If you use the **rtmp-filtering | no-rtmp-filtering** parameter, you are configuring an AppleTalk network filter. See the following section.

Appletalk Network Filters

Routing Table Maintenance Protocol (RTMP) filtering enhances a zone filter by hiding the cable ranges inside the zones used by other routers. The denied network numbers of the filtered zone will be removed from the RTMP packets.

The Macintosh chooser uses ZIP GetZoneList request to compile a list of zones available, so if the zone is not there the Macintosh computer cannot access it. RTMP filtering is useful for preventing downstream and adjacent routers from responding to GetZoneList requests that could give access to the zones you want to filter. All routers on the same segment should be configured with the same filters. You can prevent local Macintosh computers from accessing a zone but still allow the downstream routers with Macintosh computers attached to other networks to access those zones. To do so, do not use the RTMP filtering option with the zone filter.

When you configure an AppleTalk zone filter to also filter network information, the Foundry device removes route information for the networks in the specified zone before sending the RTMP packet out on the port.

Actions

AppleTalk network filters remove information about the networks in the denied zones before sending RTMP packets to Macintosh computers.

NOTE: AppleTalk network filters only deny information; they do not permit information.

Scope

You configure and apply AppleTalk network filters on individual ports.

Syntax

Use the following CLI commands or Web management interface panels to configure AppleTalk zone filters.

Table B.21: AppleTalk Zone Filters

CLI syntax	Web management links
BigIron(config-if-1/1)# appletalk permit zone <string>	Configure->AppleTalk->Zone Filter
BigIron(config-if-1/1)# appletalk deny zone <string> | additional-zones rtmp-filtering | no-rtmp-filtering	Configure->AppleTalk->Additional Zone Filter

NOTE: If you do not use the **rtmp-filtering | no-rtmp-filtering** parameter, you are configuring an AppleTalk zone filter.

BGP4 Filters

Border Gateway Protocol version 4 (BGP4) filters control the routes that a Foundry device learns from BGP4 neighbors and advertises to BGP4 neighbors. You can configure filters to filter route information based on network address, AS-path, or community name.

BGP4 Address Filters

BGP4 address filters control whether the Foundry device learns or drops BGP4 route information based on the route's network address.

Actions

- A BGP4 address filter applied to inbound packets permits (learns) or denies (drops) the specified network address in BGP4 updates received from a BGP4 neighbor.
- A BGP4 address filter applied to outbound packets permits (advertises) or denies (drops) the specified network address in BGP4 updates the Foundry device sends to a BGP4 neighbor.

Scope

You define BGP4 address filters globally, then apply them as part of a BGP4 neighbor's distribute list or as part of a match statement in a route map.

Syntax

Use the following CLI commands or Web management interface panels to configure BGP4 address filters.

Table B.22: BGP4 Address Filters

CLI syntax	Web management links
BigIron(config-bgp-router)# address-filter <num> permit | deny <ip-addr> <ip-mask> | any <ip-addr> <ip-mask> | any	Configure->BGP->Address Filter
BigIron(config-bgp-router)# neighbor <router-id> remote-as <as-number> [advertisement-interval <num>] [distribute-list in | out <num,num,...>] [ebgp-multihop] [filter-list in | out <num,num,...>] [maximum-prefix <num>] [next-hop-self] [remote-as <as-number>] [route-map <map-name>] [send-community] [weight <num>]	Configure->BGP->Neighbor
BigIron(config-bgp-routemap RMAP_NAME)# match as-path-filters | community-filters | address-filters <num,num,...> [metric <num>] [next-hop <ip-addr>] [route-type internal | external-type1 | external-type2] [tag <tag-value>]	Configure->BGP->Route Map Filter

NOTE: The **neighbor** command adds a BGP neighbor. The **distribute-list** parameter specifies a list of address filters and whether the list is applied to inbound or outbound BGP updates.

NOTE: The **match** command compares the information you configure for the command's parameters against BGP routes. You use this command when configuring a route map. If the comparison matches a route, set statements in the route map specify the action to take. See "Defining Route Maps" on page 16-73.

BGP4 AS-Path Filters

BGP4 AS-path filters control whether the Foundry device learns or drops BGP4 route information based on the route's AS-path. The ***AS-path*** is the list of BGP4 autonomous systems (ASs) through which the route information has traveled to reach the Foundry device.

Actions

- A BGP4 AS-path filter applied to inbound packets permits (learns) or denies (drops) routes for networks with the specified AS-path in BGP4 updates received from a BGP4 neighbor.
- A BGP4 AS-path filter applied to outbound packets permits (advertises) or denies (drops) routes for networks with the specified AS-path in BGP4 updates sent to a BGP4 neighbor.

Scope

You define BGP4 AS-path filters globally, then apply them as part of a BGP4 neighbor's distribute list or as part of a match statement in a route map.

Syntax

Use the following CLI commands or Web management interface panels to configure BGP4 AS-path filters.

Table B.23: BGP4 AS-Path Filters

CLI syntax	Web management links
BigIron(config-bgp-router)# as-path-filter <num> permit | deny <as-path>	Configure->BGP->AS Path Filter
BigIron(config-bgp-router)# neighbor <router-id> remote-as <as-number> [advertisement-interval <num>] [distribute-list in | out <num,num,...>] [ebgp-multihop] [filter-list in | out <num,num,...>] [maximum-prefix <num>] [next-hop-self] [remote-as <as-number>] [route-map <map-name>] [send-community] [weight <num>]	Configure->BGP->Neighbor
BigIron(config-bgp-routemap RMAP_NAME)# match as-path-filters | community-filters | address-filters <num,num,...> [metric <num>] [next-hop <ip-addr>] [route-type internal | external-type1 | external-type2] [tag <tag-value>]	Configure->BGP->Route Map Filter

NOTE: The <as-path> value can be a regular expression. See "Using Regular Expressions" on page 16-63.

NOTE: The **neighbor** command adds a BGP neighbor. The **filter-list** parameter specifies a list of AS-path filters and whether the list is applied to inbound or outbound BGP updates.

NOTE: The **match** command compares the information you configure for the command's parameters against BGP routes. You use this command when configuring a route map. If the comparison matches a route, set statements in the route map specify the action to take. See "Defining Route Maps" on page 16-73.

BGP4 Community Filters

BGP4 community filters control whether the Foundry device learns or drops BGP4 route information based on the route's community membership.

Actions

- A BGP4 community filter applied to inbound packets permits (learns) or denies (drops) routes for networks with the specified community membership in BGP4 updates received from a BGP4 neighbor.
- A BGP4 AS-path filter applied to outbound packets permits (advertises) or denies (drops) routes for networks with the specified community membership in BGP4 updates sent to a BGP4 neighbor.

Scope

You define BGP4 community filters globally, then apply them as part of a BGP4 neighbor's distribute list or as part of a match statement in a route map.

Syntax

Use the following CLI commands or Web management interface panels to configure BGP4 community filters.

Table B.24: BGP4 Community Filters

CLI syntax	Web management links
BigIron(config-bgp-router)# community-filter <filter-num> permit | deny <num> | internet | no-advertise | no-export	Configure->BGP->Community Filter
BigIron(config-bgp-routemap RMAP_NAME)# match as-path-filters | community-filters | address-filters <num,num,...> [metric <num>] [next-hop <ip-addr>] [route-type internal | external-type1 | external-type2] [tag <tag-value>]	Configure->BGP->Route Map Filter

NOTE: The **match** command compares the information you configure for the command's parameters against BGP routes. You use this command when configuring a route map. If the comparison matches a route, set statements in the route map specify the action to take. See "Defining Route Maps" on page 16-73.

Redistribution Filters

Redistribution filters control the exchange of routes between routing protocols. RIP, OSPF, and BGP4 support redistribution of one another's routes. In addition, they all allow exchange of static routes.

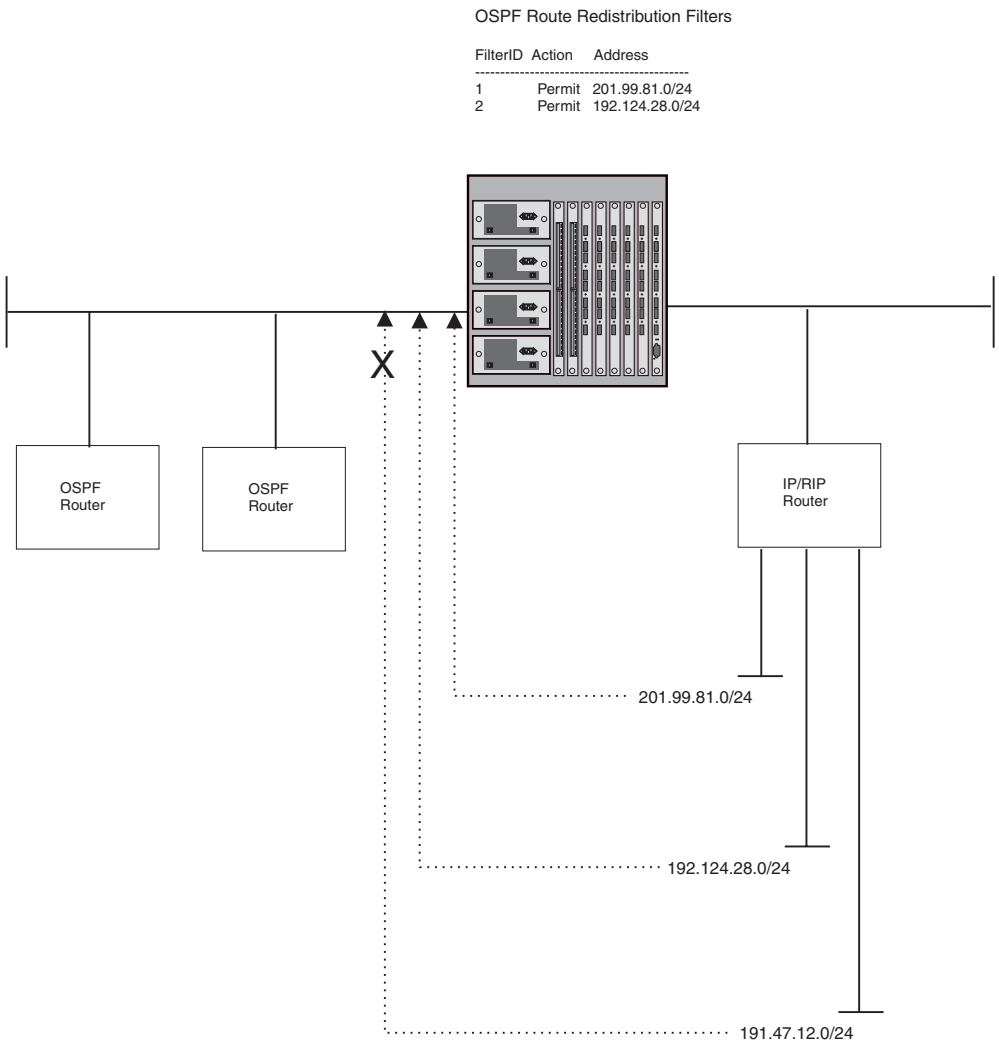
You configure RIP and OSPF redistribution filters to permit or deny routes for specific network addresses. Optionally, you can also filter on and modify the route metric. To configure redistribution, you configure redistribution filters in the protocol that will receive the routes. Redistribution is disabled by default in RIP and OSPF and enabled by default in BGP4.

BGP4 redistribution filters can filter based on a route's metric, weight, and also on the results of comparison of the route information with a route map. A ***route map*** is a named set of match conditions and parameter settings that a Foundry Layer 3 Switch can use to modify route attributes and to control redistribution of routes. For more information, see "Defining Route Maps" on page 16-73.

BGP4 allows you to include the redistribution filters as part of a route map. A route map examines and modifies route information exchanged between BGP4 and RIP or OSPF. See "Configuring BGP4" on page 16-1 for more information.

Figure B.10 shows an example of a redistribution filter. In this example, redistribution filters in OSPF are configured to redistribute two RIP routes into OSPF. Notice that unlike some other filter examples in this appendix, a filter for permitting all routes (to change the default action) is not configured. The default redistribution action is permit, even after you configure a redistribution filter. To maintain tight control over redistribution, define a "deny any" redistribution filter as the last filter (the one with the highest ID) and deny permit filters for specific routes.

Figure B.10 OSPF redistribution filters



RIP Redistribution Filters

RIP redistribution filters control redistribution of routes from other protocols into RIP. A Foundry device running RIP can redistribute static routes, OSPF routes, and BGP4 routes (if BGP4 is supported on the device) into RIP.

Optionally, you can specify a metric that the route must match or you can set the metric on redistributed routes. By setting the metric, you can cause the router to prefer RIP routes or redistributed routes to the specified network.

Actions

RIP redistribution filters permit (redistribute) or deny (do not redistribute) OSPF or BGP4 routes into RIP.

Scope

You configure RIP redistribution filters globally. They are automatically applied as soon as you configure them.

Syntax

Use the following CLI commands or Web management interface panels to configure RIP redistribution filters.

Table B.25: RIP Redistribution Filters

CLI syntax	Web management links
BigIron(config-rip-router)# permit | deny redistribute <filter-num> all | bgp | ospf | static address <ip-addr> <ip-mask> [match-metric <value> | set-metric <value>]	Configure->RIP->Redistribution Filter

OSPF Redistribution Filters

OSPF redistribution filters control redistribution of routes from other protocols into OSPF. A Foundry device running OSPF can redistribute static routes, RIP routes, and BG4P routes (if BGP4 is supported on the device) into OSPF.

Optionally, you can specify a metric that the route must match or you can set the metric on redistributed routes. By setting the metric, you can cause the router to prefer OSPF routes or redistributed routes to the specified network.

Actions

OSPF redistribution filters permit (redistribute) or deny (don't redistribute) RIP or BGP4 routes into OSPF.

Scope

You configure and apply OSPF redistribution filters globally.

Syntax

Use the following CLI commands or Web management interface panels to configure OSPF redistribution filters.

Table B.26: OSPF Redistribution Filters

CLI syntax	Web management links
BigIron(config-ospf-router)# deny | permit redistribute <filter-num> all | bgp | rip | static address <ip-addr> [match-metric <value> | set-metric <value>]	Configure->OSPF->Redistribution Filter

BGP4 Redistribution Filters

BGP4 redistribution filters control redistribution of routes from other protocols into BGP4. A Foundry device running BGP4 can redistribute static routes, RIP routes, and OSPF routes into BGP4.

Optionally, you can modify a route's metric and weight and use a route map to change additional attributes of the route.

Actions

BGP4 redistribution filters permit (redistribute) or deny (don't redistribute) RIP or OSPF routes into RIP.

Scope

You configure and apply BGP4 redistribution filters globally.

Syntax

Use the following CLI commands or Web management interface panels to configure BGP4 redistribution filters.

Table B.27: BGP4 Redistribution Filters

CLI syntax	Web management links
BigIron(config-bgp-router)# redistribute rip | ospf | static [match internal | external1 | external2] [metric <num>] [route-map <name>] [weight <num>]	Configure->BGP->Redistribute

NOTE: The optional **match internal | external1 | external2** argument applies only to OSPF.

IP Switching and IPX Switching Filters

IP and IPX switching filters are the same as router acceleration policies. See “Router Acceleration Policies” on page B-18.

Layer 4 Filters

Layer 4 filters control IP traffic based on the Layer 3 and Layer 4 information in the packets. On Foundry Layer 2 Switches and Layer 3 Switches, Layer 4 filters are access policies that control access to Layer 4 applications based on TCP/UDP or other port number.

On the ServerIron, Layer 4 filters are policies that control whether the ServerIron redirects HTTP traffic from web clients to cache servers or sends the traffic to the Internet.

TCP/UDP Forwarding Filters

TCP/UDP forwarding filters are the same as TCP/UDP access policies. See “TCP/UDP Access Policies” on page B-20.

Cache Server Redirection Filters

Cache server redirection filters are the same thing as cache server redirection policies. See “Cache Server Redirection Policies” on page B-22.

