kontron

»Kontron User's Guide«

FASTPATH
Configuration Guide

Document Revision 1.1

Document ID: FASTPATH Configuration Guide

Issue Date: June 2011

Revision History

Rev. Index	Brief Description of Changes	Date of Issue
1.0	Initial Issue	13.05.2011
1.1	Minor edits in all chapters	07.06.2011

Customer Service

Contact Information:

Kontron Canada, Inc.
4555 Ambroise-Lafortune
Boisbriand, Québec, Canada
J7H 0A4
Tel: (450) 437-5682
(800) 354-4223
Fax: (450) 437-8053
E-mail: support@ca.kontron.com

Kontron Modular Computer GmbH
Sudetenstrasse 7
87600 Kaufbeuren
Germany
+49 (0) 8341 803 333
+49 (0) 8341 803 339
support-kom@kontron.com

Visit our site at:www.kontron.com

Table of Contents

Revision History ..... *ii*
Customer Service ..... *ii*
Proprietary Note ..... *vii*
Trademarks ..... *vii*
About This Document ..... *vii*
Kontron Support for Switch Software ..... *viii*
Audience ..... *viii*
Organization ..... *ix*
Additional Documentation ..... *ix*
Advisory Conventions ..... *ix*
Typographical Conventions ..... *x*
About Fastpath Software Modules ..... *x*
Two Year Warranty ..... *xi*

1. System Configuration ***1***
1.1 Traceroute *1*
1.1.1 CLI Example *2*
1.2 Configuration Scripting *2*
1.2.1 Overview *2*
1.2.2 Considerations *3*
1.2.3 CLI Examples *3*
1.3 Outbound Telnet *5*
1.3.1 Overview *5*
1.3.2 CLI Examples *6*
1.4 Pre-Login Banner *7*
1.4.1 Overview *7*
1.4.2 CLI Example *7*
1.5 Simple Network Time Protocol (SNTP) *8*
1.5.1 Overview *8*
1.5.2 CLI Examples *8*
1.6 Syslog *10*
1.6.1 Overview *10*
1.6.2 CLI Examples *10*
1.7 Port Description *13*
1.7.1 CLI Example *13*
1.8 Storm Control *13*
1.8.1 CLI Example *14*
1.9 Cable Test *14*
1.9.1 CLI Example *15*

2. Switching Configuration ***16***
2.1 Virtual LANs *16*
2.1.1 VLAN Configuration Example *17*
2.1.2 CLI Examples *17*
2.1.3 Private Edge VLANs *19*
2.1.4 CLI Example *19*
2.2 IGMP Snooping *20*
2.2.1 Overview *20*

2.2.2 CLI Examples 20
2.3 IGMP Proxy 22
2.3.1 CLI examples 22
2.4 Link Aggregation/Port-channels 23
2.4.1 CLI Example 23
2.5 Port Mirroring 25
2.5.1 Overview 25
2.5.2 CLI Examples 26
2.6 Port Security 27
2.6.1 Overview 27
2.6.2 Operation 28
2.6.3 CLI Examples 28
2.7 Link Layer Discovery Protocol 29
2.7.1 CLI Examples 29
2.8 Denial of Service Attack Protection 30
2.8.1 Overview 30
2.8.2 CLI Examples 31
2.9 DHCP Filtering 31
2.9.1 Overview 31
2.9.2 Limitations 31
2.9.3 CLI Examples 32
2.10Configuring Spanning Tree Protocol 33
2.10.1 Configuring Spanning Tree Protocol 33

3. Routing Configuration 38
3.1 Port Routing 38
3.1.1 Port Routing Configuration 38
3.2 VLAN Routing 40
3.2.1 CLI Examples 40
3.2.2 VLAN Routing RIP Configuration 42
3.2.3 VLAN Routing OSPF Configuration 45
3.3 Virtual Router Redundancy Protocol 46
3.3.1 CLI Examples 47
3.4 Proxy Address Resolution Protocol (ARP) 49
3.4.1 Overview 49
3.4.2 CLI Examples 49
3.5 OSPF 50
3.5.1 OSPF Concepts and Terms 50
3.5.2 CLI Examples 51
3.6 Routing Information Protocol 58
3.6.1 RIP Configuration 58
3.6.2 CLI Examples 59
3.7 Route Preferences 60
3.7.1 Assigning Administrative Preferences to Routing Protocols 60
3.7.2 Assigning Administrative Preferences to Static Routes 61
3.7.3 Using Equal Cost Multipath 62
3.8 Loopback Interfaces 63

4. Device Security 65
4.1 802.1x Network Access Control 65
4.1.1 802.1x Network Access Control Example 66
4.2 Access Control Lists (ACLs) 67
4.2.1 Overview 67

4.2.2 MAC ACLs 68
4.2.3 IP ACLs 68
4.2.4 ACL Configuration Process 69
4.2.5 IP ACL CLI Examples 69
4.2.6 MAC ACL CLI Examples 70
4.3 RADIUS 73
4.3.1 RADIUS Configuration Example 73
4.4 TACACS+ 75
4.4.1 TACACS+ Configuration Example 75

5. IPv6 77
5.1 Overview 77
5.2 Interface Configuration 77
5.2.1 CLI Example 78
5.3 DHCPv6 80
5.3.1 CLI Examples 81

6. Quality of Service 82
6.1 Class of Service Queuing 82
6.1.1 Ingress Port Configuration 82
6.1.2 Egress Port Configuration—Traffic Shaping 83
6.1.3 Queue configuration 83
6.1.4 Queue Management Type 83
6.1.5 CLI Examples 83
6.2 Differentiated Services 86
6.2.1 CLI Example 87
6.2.2 DiffServ for VoIP Configuration Example 89

7. Multicast 91
7.1 Overview 91
7.2 IGMP Configuration 91
7.2.1 CLI Example 92
7.3 IGMP Proxy 92
7.3.1 CLI Examples 92
7.4 MLD 94
7.4.1 CLI Example 94
7.5 DVMRP 94
7.5.1 CLI Example 95
7.6 PIM 96
7.6.1 PIM-SM 96
7.6.2 PIM-DM 98

List of Figures

Figure 1-1: Log Files Key 10
Figure 2-1: VLAN Example Network Diagram 17
Figure 2-2: LAG/Port-channel Example Network Diagram 23
Figure 3-1: Port Routing Example Network Diagram 38
Figure 3-2: VLAN Routing Example Network Diagram 40
Figure 3-3: RIP for VLAN Routing Example Network Diagram 42
Figure 3-4: VRRP Example Network Configuration 46
Figure 3-5: OSPF Example Network Diagram: Border Router 51
Figure 3-6: OSPF Configuration—Stub Area and NSSA Area 53
Figure 3-7: OSPF Configuration—Virtual Link 55
Figure 3-8: Port Routing Example Network Diagram 58
Figure 3-9: Forwarding Without ECMP 61
Figure 3-10:Next Hop with Two Static Routes 61
Figure 4-1: FASTPATH with 802.1x Network Access Control 64
Figure 4-2: IP ACL Example Network Diagram 67
Figure 4-3: RADIUS Servers in a FASTPATH Network 73
Figure 4-4: FASTPATH with TACACS+ 74
Figure 5-1: IPv6 Example 77
Figure 5-2: DHCPv6 Prefix Delegation Scenario 79
Figure 6-1: CoS Mapping and Queue Configuration 83
Figure 6-2: CoS Configuration Example System Diagram 84
Figure 6-3: DiffServ Internet Access Example Network Diagram 86
Figure 6-4: DiffServ VoIP Example Network Diagram 88

Proprietary Note

Trademarks

About This Document

This configuration guide provides examples of the use of FASTPATH software in typical network applications. It describes the use and advantages of specific functions provided by FASTPATH software, and includes information on configuring those functions using the command line interface (CLI).

The Configuration Guide is relevant for the following Kontron product families:

- CompactPCI Switches CP6930, CP6923 and CP3923
- MicroTCA Carrier Hubs AM4904 and AM4910
- AdvancedTCA Carrier AT8404
- AdvancedTCA Switches AT8904 and AT8910
- VPX Switch VX3910

CLI commands and configuration options may vary depending on the particular product and the FASTPATH version running. There may be differences in command syntax or command availability, please refer to the CLI Reference Manual provided for the product you are using.

Kontron Support for Switch Software

In case of support questions related to the Fastpath software on any of the products, please contact Kontron Support. Contact details are given in the corresponding product User's Guide.

To be able to process support cases as fast as possible, please add the following information:

- Output of
  - show boardinfo version
  - show tech-support
- Information of use-case
  - Overall system setup
  - Block diagram of used I/Fs and connected devices
  - Configuration of external devices (ETHx setup, ...)

Audience

The information in this guide is intended for any of the following individuals:

- System administrators who are responsible for configuring and operating a network using FASTPATH software
- Software engineers who are integrating FASTPATH software into a router or switch product
- Level 1 and/or Level 2 Support providers

To obtain the greatest benefit from this guide, you should have an understanding of the base software and should have read the specification for your networking device platform. You should also have basic knowledge of Ethernet and networking concepts.

Organization

This document is organized as follows:

- Chapter 1: “System Configuration” on page 1 describes how to configure basic system and port settings, use system interfaces and utilities, and create and use CLI scripts.
- Chapter 2: “Switching Configuration” on page 16 provides configuration scenarios for switching, including creating virtual LANs, IGMP snooping interfaces, STP and port security.
- Chapter 3: “Routing Configuration” on page 38 provides configuration scenarios for layer 3 features such as VLAN routing, OSPF, and RIP.
- Chapter 4: “Device Security” on page 65 provides information on creating access control lists and configuring RADIUS and TACACS+ servers.
- Chapter 5: “IPv6” on page 77 describes configuring and using IPv6-enabled interfaces in a mixed IPv6/IPv4 network.
- Chapter 6: “Quality of Service” on page 82 provides configuration scenarios for class-of-service queueing and differentiated services.
- Chapter 7: “Multicast” on page 91 describes IGMP, MLD, DVMRP and PIM.

Additional Documentation

The following documentation provides additional information about FASTPATH software:

- The FASTPATH CLI Reference manual of the corresponding product describes the commands available from the command-line interface (CLI) for managing, monitoring, and configuring the switch.

Advisory Conventions

This section describes the conventions this document uses.

CAUTION

This symbol provides information about critical aspects of the configuration, combinations of settings, events or procedures that can adversely affect network connectivity, security and so on.

Note...

This symbol and title emphasize aspects the reader should read through carefully for his or her own advantage.

Typographical Conventions

This guide uses the typographical conventions described in the table below.

Symbol	Description	Example
Blue Text	Hyperlinked text.	See "About This Document"on page 1
`courier font`	Command-line text and file names	`(switch-prompt)#`

About Fastpath Software Modules

The FASTPATH software suite includes the following modules:

- Switching (Layer 2)
- Routing (Layer 3)
- IPv6 routing
- Multicast
- Quality of Service
- Management (CLI and SNMP)

Not all modules are available for all platforms or software releases.

Two Year Warranty

Kontron AG grants the original purchaser of Kontron's products a *TWO YEAR LIMITED HARDWARE WARRANTY* as described in the following. However, no other warranties that may be granted or implied by anyone on behalf of Kontron are valid unless the consumer has the express written consent of Kontron AG.

Kontron AG warrants their own products, excluding software, to be free from manufacturing and material defects for a period of 24 consecutive months from the date of purchase. This warranty is not transferable nor extendible to cover any other users or long-term storage of the product. It does not cover products which have been modified, altered or repaired by any other party than Kontron Modular Computers GmbH or their authorized agents. Furthermore, any product which has been, or is suspected of being damaged as a result of negligence, improper use, incorrect handling, servicing or maintenance, or which has been damaged as a result of excessive current/voltage or temperature, or which has had its serial number(s), any other markings or parts thereof altered, defaced or removed will also be excluded from this warranty.

If the customer's eligibility for warranty has not been voided, in the event of any claim, he may return the product at the earliest possible convenience to the original place of purchase, together with a copy of the original document of purchase, a full description of the application the product is used on and a description of the defect. Pack the product in such a way as to ensure safe transportation (see our safety instructions).

Kontron provides for repair or replacement of any part, assembly or sub-assembly at their own discretion, or to refund the original cost of purchase, if appropriate. In the event of repair, refunding or replacement of any part, the ownership of the removed or replaced parts reverts to Kontron Modular Computers GmbH, and the remaining part of the original guarantee, or any new guarantee to cover the repaired or replaced items, will be transferred to cover the new or repaired items. Any extensions to the original guarantee are considered gestures of goodwill, and will be defined in the "Repair Report" issued by Kontron with the repaired or replaced item.

Kontron Modular Computers GmbH will not accept liability for any further claims resulting directly or indirectly from any warranty claim, other than the above specified repair, replacement or refunding. In particular, all claims for damage to any system or process in which the product was employed, or any loss incurred as a result of the product not functioning at any given time, are excluded. The extent of Kontron Modular Computers GmbH liability to the customer shall not exceed the original purchase price of the item for which the claim exists.

Kontron Modular Computers GmbH issues no warranty or representation, either explicit or implicit, with respect to its products' reliability, fitness, quality, marketability or ability to fulfil any particular application or purpose. As a result, the products are sold "as is," and the responsibility to ensure their suitability for any given task remains that of the purchaser. In no event will Kontron be liable for direct, indirect or consequential damages resulting from the use of our hardware or software products, or documentation, even if Kontron were advised of the possibility of such claims prior to the purchase of the product or during any period since the date of its purchase.

Please remember that no Kontron Modular Computers GmbH employee, dealer or agent is authorized to make any modification or addition to the above specified terms, either verbally or in any other form, written or electronically transmitted, without the company's consent.

1. System Configuration

This chapter provides configuration scenarios for the following features:

- Traceroute
- Configuration Scripting
- Outbound Telnet
- Pre-Login Banner
- Simple Network Time Protocol (SNTP)
- Syslog
- Port Description
- Storm Control
- Cable Test

Note...

For information on setting up the hardware and serial or TFTP connection, refer to the product User Guide.

1.1 Traceroute

Use Traceroute to discover the routes that packets take when traveling on a hop-by-hop basis to their destination through the network.

- Maps network routes by sending packets with small Time-to-Live (TTL) values and watches the ICMP time-out announcements
- Command displays all L3 devices
- Can be used to detect issues on the network
- Tracks up to 20 hops
- Default UPD port uses 33343 unless modified in the traceroute command

1.1.1 CLI Example

The following shows an example of using the traceroute command to determine how many hops there are to the destination. The command output shows each IP address the packet passes through and how long it takes to get there. In this example, the packet takes 16 hops to reach its destination.

```
(Ethernet Fabric) #traceroute ?
<ipaddr>            Enter IP address.

(Ethernet Fabric) #traceroute 216.109.118.74 ?
cr>                 Press enter to execute the command.
count               Number of probes per hop.
initTtl             Initial TTL to be used.
interval            Time between probes in seconds.
maxFail             Max failures allowed in session
maxTtl              Maximum TTL for the destination.
port                UDP Dest port in probe packets.
size                Size of probe packets.

(Ethernet Fabric) #traceroute 216.109.118.74

Tracing route over a maximum of 20 hops

 1  10.254.24.1         40 ms        9 ms       10 ms
 2  10.254.253.1        30 ms       49 ms       21 ms
 3  63.237.23.33        29 ms       10 ms       10 ms
 4  63.144.4.1          39 ms       63 ms       67 ms
 5  63.144.1.141        70 ms       50 ms       50 ms
 6  205.171.21.89       39 ms       70 ms       50 ms
 7  205.171.8.154       70 ms       50 ms       70 ms
 8  205.171.8.222       70 ms       50 ms       80 ms
 9  205.171.251.34      60 ms       90 ms       50 ms
10  209.244.219.181     60 ms       70 ms       70 ms
11  209.244.11.9        60 ms       60 ms       50 ms
12  4.68.121.146        50 ms       70 ms       60 ms
13  4.79.228.2          60 ms       60 ms       60 ms
14  216.115.96.185     110 ms       59 ms       70 ms
15  216.109.120.203     70 ms       66 ms       95 ms
16  216.109.118.74      78 ms      121 ms       69 ms
```

1.2 Configuration Scripting

Configuration scripting allows you to generate a text-formatted script file that shows the current configuration of the system. You can generate multiple scripts and upload and apply them to more than one switch.

1.2.1 Overview

Configuration scripting:

- Provides scripts that can be uploaded and downloaded to the system.
- Provides flexibility to create command configuration scripts.

- Can be applied to several switches.
- Can save up to ten scripts or 500K of memory.
- Provides List, Delete, Apply, Upload, Download.
- Provides script format of one CLI command per line.

1.2.2 Considerations

- Total number of scripts stored on the system is limited by NVRAM/FLASH size.
- Application of scripts is partial if script fails. For example, if the script executes five of ten commands and the script fails, the script stops at five.
- Scripts cannot be modified or deleted while being applied.
- Validation of scripts checks for syntax errors only. It does not validate that the script will run.

1.2.3 CLI Examples

The following are examples of the commands used for configurations scripting.

1.2.3.1 Example #1: script

```
(Ethernet Fabric) #script ?

apply         Applies configuration script to the switch.
delete        Deletes a configuration script file from the switch.
list          Lists all configuration script files present on the switch.
show          Displays the contents of configuration script.
validate      Validate the commands of configuration script.
```

1.2.3.2 Example #2: script list and script delete

```
(Ethernet Fabric) #script list

Configuration Script Name     Size(Bytes)
--------------------------    -----------
basic.scr                           93
running-config.scr                3201

2 configuration script(s) found.
1020706 bytes free.

(Ethernet Fabric) #script delete basic.scr

Are you sure you want to delete the configuration script(s)? (y/n) y

1 configuration script(s) deleted.
```

1.2.3.3 Example #3: script apply running-config.scr

```
(Ethernet Fabric) #script apply running-config.scr

Are you sure you want to apply the configuration script? (y/n) y

The systems has unsaved changes.
Would you like to save them now? (y/n) y

Configuration Saved!
```

1.2.3.4 Example #4: show running-config

Use this command to capture the running configuration into a script.

```
(Ethernet Fabric) #show running-config running-config.scr

Config script created successfully.

(Ethernet Fabric) #script list

Configuration Script NameSize(Bytes)
-----------------------------------
running-config.scr            3201

1 configuration script(s) found.
1020799 bytes free.
```

1.2.3.5 Example #5: copy nvram: script

Use this command to upload a configuration script.

```
(Ethernet Fabric) #copy nvram:script running-config.scr
tftp://192.168.77.52/running-config.scr

Mode.........................TFTP
Set TFTP Server IP...........192.168.77.52
TFTP Path....................../
TFTP Filename................running-config.scr
Data Type....................Config Script
Source Filename..............running-config.scr

Are you sure you want to start? (y/n) y

File transfer operation completed successfully.
```

1.2.3.6 Example #6: script validate running-config.scr

```
(Ethernet Fabric) #script validate running-config.scr
serviceport protocol none
network protocol dhcp
no network javamode
vlan database
exit
configure
stack
member 2 1
exit
logging buffered
logging host 192.168.77.151
```

```
Configuration script 'running-config.scr' validated.

(Ethernet Fabric) #script apply running-config.scr

Are you sure you want to apply the configuration script? (y/n) y
The system has unsaved changes.
Would you like to save them now? (y/n) y
Configuration Saved!
```

1.2.3.7 Example #7: Validate another Configuration Script

```
(Ethernet Fabric) #script validate default.scr

network parms 172.30.4.2 255.255.255.0 0.0.0.0
vlan database
exit
configure
lineconfig
exit
spanning-tree configuration name 00-18-00-00-00-10
interface 0/1
exit
interface 0/2
exit
interface 0/3
exit
... continues through interface 0/26 ...
exit
exit
Configuration script 'default.scr' validation succeeded.
```

1.3 Outbound Telnet

1.3.1 Overview

Outbound telnet:

- Establishes an outbound telnet connection between a device and a remote host.
- When a telnet connection is initiated, each side of the connection is assumed to originate and terminate at a "Network Virtual Terminal" (NVT).
- Server and user hosts do not maintain information about the characteristics of each other's terminals and terminal handling conventions.
- Must use a valid IP address.

1.3.2 CLI Examples

The following are examples of the commands used in the outbound telnet feature.

1.3.2.1 Example #1: show network

```
(Ethernet Fabric) >telnet 192.168.77.151
Trying 192.168.77.151...
(Ethernet Fabric)
User:admin
Password:
(Ethernet Fabric) >enable
Password:

(Ethernet Fabric) #show network

IP Address.............................192.168.77.151
Subnet Mask............................255.255.255.0
Default Gateway..........................  192.168.77.127
Burned In MAC Address.....................  00:10:18.82.04:E9
Locally Administered MAC Address.........00:00:00:00:00:00
MAC Address Type.........................Burned In
Network Configuration Protocol Current...DHCP
Management VLAN ID.......................  1
Web Mode.................................  Enable
Java Mode ...............................  Disable
```

1.3.2.2 Example #2: show telnet

```
(Ethernet Fabric) #show telnet

Outbound Telnet Login Timeout (minutes)....... 5
Maximum Number of Outbound Telnet Sessions.....5
Allow New Outbound Telnet Sessions.............Yes
```

1.3.2.3 Example #3: transport output telnet

```
(Ethernet Fabric) (config) #line ?

console                   Enter into Line Console Config Mode.
ssh                       Enter into Line SSH Config Mode.
telnet                    Enter into Line Telnet Config Mode.

Ethernet Fabric) (config) #line console

(Ethernet Fabric) (Line-config) #transport ?

input                     Displays the protocols to use to connect to a
                          specific line of the router.
output                    Displays the protocols to use for outgoing
                          connections from a line.

(Ethernet Fabric) (Line-config) #transport output ?

telnet                    Allow or disallow new telnet sessions.

(Ethernet Fabric) (Line-config) #transport output telnet ?

<cr>                      Press Enter to execute the command.

(Ethernet Fabric) (Line-config) #transport output telnet
```

1.3.2.4 Example #4: session-limit and session-timeout

```
(Ethernet Fabric) (Line-config)#session-limit ?

<0-5>                    Configure the maximum number of outbound telnet
                         sessions allowed.

(Ethernet Fabric) (Line-config)#session-limit 5

(Ethernet Fabric) (Line-config)#session-timeout ?

<1-160>                  Enter time in minutes.

(Ethernet Fabric) (Line-config)#session-timeout 15
```

1.4 Pre-Login Banner

1.4.1 Overview

The Pre-Login Banner feature allows you to create message screens when logging into the CLI Interface. By default, no Banner file exists. A banner up to size 2K can be uploaded or downloaded.

The Pre-Login Banner feature is only for the CLI interface.

1.4.2 CLI Example

To create a Pre-Login Banner, follow these steps:

1. On your PC, using Notepad or another text editor, create a banner.txt file that contains the banner to be displayed.

```
FASTPATH's Login Banner - Unauthorized access is punishable by law.
```

2. Transfer the file from the PC to the switch using TFTP

```
(Ethernet Fabric) #copy tftp://192.168.77.52/banner.txt nvram:clibanner

Mode..................................... TFTP
Set TFTP Server IP.............................192.168.77.52
TFTP Path......................................./
TFTP Filename..................................banner.txt
Data Type.....................................Cli Banner

Are you sure you want to start? (y/n) y
CLI Banner file transfer operation completed successfully!
exit
(Ethernet Fabric) >logout
FASTPATH's Login Banner - Unauthorized access is punishable by law.
User:
```

Note...

The command "no clibanner" removes the banner from the switch.

1.5 Simple Network Time Protocol (SNTP)

1.5.1 Overview

The SNTP implementation has the following features:

- Used for synchronizing network resources
- Adaptation of NTP
- Provides synchronized network timestamp
- Can be used in broadcast or unicast mode
- SNTP client implemented over UDP that listens on port 123

1.5.2 CLI Examples

The following are examples of the commands used in the SNTP feature.

1.5.2.1 Example #1: show sntp

```
(Ethernet Fabric) #show sntp ?

<cr> Press Enter to execute the command.
clientDisplay SNTP Client Information.
serverDisplay SNTP Server Information.
```

1.5.2.2 Example #2: show sntp client

```
(Ethernet Fabric) #show sntp client

Client Supported Modes:unicast broadcast
SNTP Version:4
Port:123
Client Mode:unicast
Unicast Poll Interval:6
Poll Timeout (seconds):5
Poll Retry:1
```

1.5.2.3 Example #3: show sntp server

```
(Ethernet Fabric) #show sntp server

Server IP Address:            81.169.155.234
Server Type:                  ipv4
Server Stratum:               3
Server Reference Id:          NTP Srv: 212.186.110.32
Server Mode:                  Server
Server Maximum Entries:       3
Server Current Entries:       1

SNTP Servers
------------
```

```
IP Address:                    81.169.155.234
Address Type:                  IPV4
Priority:                      1
Version:                       4
Port:                          123
Last Update Time:              MAY 18 04:59:13 2005
Last Attempt Time:             MAY 18 11:59:33 2005
Last Update Status:            Other
Total Unicast Requests:        1111
Failed Unicast Requests:       361
```

1.5.2.4 Example #4: configure sntp

```
(Ethernet Fabric)(Config) #sntp ?

broadcast                  Configure SNTP client broadcast parameters.
client                     Configure the SNTP client parameters.
server                     Configure SNTP server parameters.
unicast                    Configure SNTP client unicast parameters.
multicast                  Configure SNTP client multicast parameters.
```

1.5.2.5 Example #5: configure sntp client mode

```
(Ethernet Fabric) (Config) #sntp client mode broadcast ?

<cr>                       Press Enter to execute the command.

(Ethernet Fabric) (Config) #sntp client mode unicast ?

<cr>                       Press Enter to execute the command.

(Ethernet FabricEthernet Fabric)(Config) #sntp broadcast client poll-interval ?

<6-10>                     Enter value in the range (6 to 10). Poll interval is 2^(value)
                           in seconds.
```

1.5.2.6 Example #6: configuring sntp server

```
(Ethernet Fabric)(Config) #sntp server Broadcom ?

<cr>                           Press Enter to execute the command.
<1-3>                          Enter SNTP server priority from 1 to 3.
```

1.5.2.7 Example #7: configure sntp client port

```
(Ethernet Fabric)(Config) #sntp client port 1 ?

<cr>                           Press Enter to execute the command.
<6-10>                         Enter value in the range (6 to 10). Poll interval is
2^(value)
                               in seconds.
```

1.6 Syslog

1.6.1 Overview

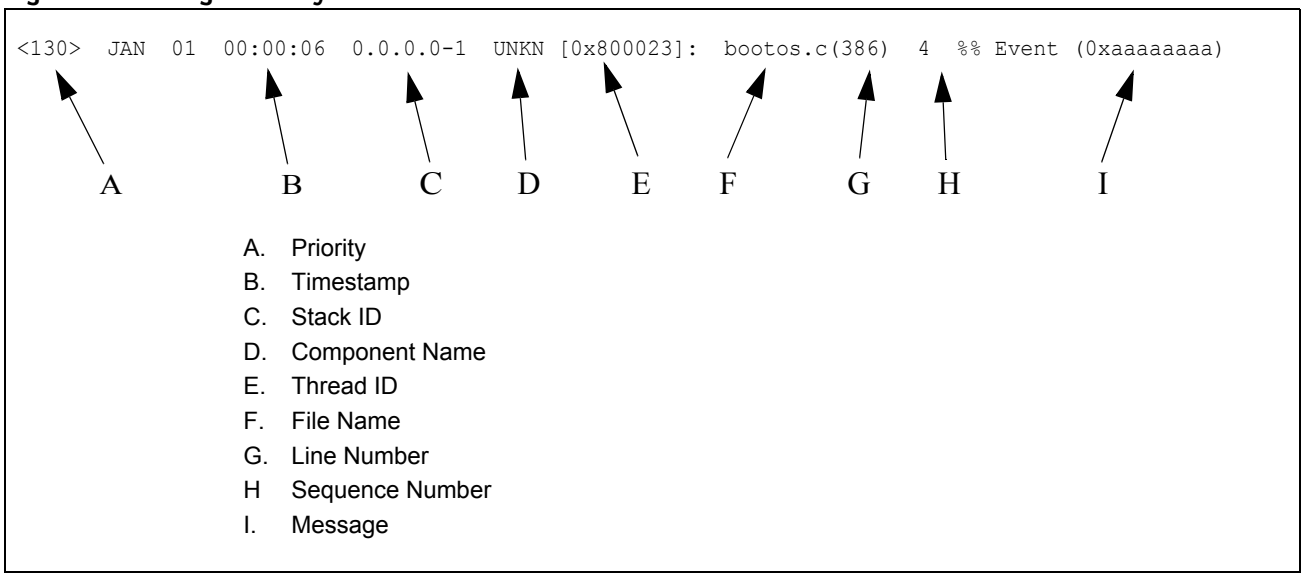
Syslog:

- Allows you to store system messages and/or errors.
- Can store to local files on the switch or a remote server running a syslog daemon.
- Provides a method of collecting message logs from many systems.

1.6.1.1 Interpreting Log Files

The figure below describes the information that displays in log messages.

Figure 1-1: Log Files Key



1.6.2 CLI Examples

The following are examples of the commands used in the Syslog feature.

1.6.2.1 Example #1: show logging

```
(Ethernet Fabric) #show logging

Logging Client Local Port         :         514
CLI Command Logging               :         disabled
Console Logging                   :         disabled
Console Logging Severity Filter   :         alert
Buffered Logging                  :         enabled

Syslog Logging                    :         enabled

Log Messages Received             :         66
Log Messages Dropped              :         0
Log Messages Relayed              :         0
```

1.6.2.2 Example #2: show logging buffered

```
(Ethernet Fabric) #show logging buffered ?

<cr>Press Enter to execute the command.

(Ethernet Fabric) #show logging buffered

Buffered (In-Memory) Logging:enabled
Buffered Logging Wrapping Behavior:On
Buffered Log Count:66


<6> Nov 29 13:31:38 0.0.0.0-1 UNKN[292290880]: sysapi.c(1280) 3 %% sysapiCfgFile
sSeparate: CRC check failed. 0x0 read and 0xce0a37e0 calculated
<6> Nov 29 13:31:38 0.0.0.0-1 UNKN[292290880]: sysapi.c(1131) 4 %% could not sep
arate SYSAPI_CONFIG_FILENAME
<2> Nov 29 13:31:42 0.0.0.0-1 UNKN[292290880]: bootos.c(332) 5 %% Event(0xaaaaaa
aa)
<6> Nov 29 13:31:49 0.0.0.0-1 UNKN[296038472]: sysapi.c(1912) 6 %% Building defa
ults for file log.cfg version 1
<6> Nov 29 13:32:12 0.0.0.0-1 UNKN[295813352]: edb.c(360) 7 %% EDB Callback: Uni
t Join: 1.
<6> Nov 29 13:32:12 0.0.0.0-1 UNKN[293358784]: sysapi.c(1912) 8 %% Building defa
ults for file simCfgData.cfg version 3
```

1.6.2.3 Example #3: show logging traplogs

```
show logging traplogs

Number of Traps Since Last Reset.............. 16
Trap Log Capacity............................. 256
Number of Traps Since Log Last Viewed......... 0

Log System Up Time           Trap
--- ----------------------- ----------------------------------------
  0 6 days 20:22:35          Failed User Login: Unit: 1 User ID:
  1 6 days 19:19:58          Multiple Users: Unit: 0 Slot: 3 Port: 1
  2 5 days 23:31:27          Multiple Users: Unit: 0 Slot: 3 Port: 1
  3 5 days 19:21:51          Multiple Users: Unit: 0 Slot: 3 Port: 1
  4 2 days 23:16:32          Link Down: Unit: 0 Slot: 1 Port: 2
  5 2 days 23:16:03          Link Down: Unit: 0 Slot: 1 Port: 1
  6 2 days 19:49:28          Multiple Users: Unit: 0 Slot: 3 Port: 1
  7 2 days 18:20:56          Multiple Users: Unit: 0 Slot: 3 Port: 1
  8 2 days 17:10:41          Multiple Users: Unit: 0 Slot: 3 Port: 1
  9 2 days 00:55:42          Multiple Users: Unit: 0 Slot: 3 Port: 1
 10 2 days 00:55:38          Failed User Login: Unit: 1 User ID: admin
 11 2 days 00:20:12          Multiple Users: Unit: 0 Slot: 3 Port: 1
```

1.6.2.4 Example #4: show logging hosts

```
(Ethernet Fabric) #show logging hosts ?

Index     IP Address       Severity   Port     Status
-----  -----------------  ----------  ----  -------------
1      192.168.21.253     critical    514   Active
```

1.6.2.5 Example #5: logging port configuration

```
(Ethernet Fabric) #config

   logging ?

buffered                  Buffered (In-Memory) Logging Configuration.
cli-command               CLI Command Logging Configuration.
console                   Console Logging Configuration.
host                      Enter IP Address for Logging Host
syslog                    Syslog Configuration.

   logging host ?

<hostaddress>             Enter Logging Host IP Address
reconfigure               Logging Host Reconfiguration
remove                    Logging Host Removal

   logging host 192.168.21.253 ?

<cr>                      Press Enter to execute the command.
<port>                    Enter Port ID from 0 to 65535

   logging host 192.168.21.253 4 ?

<cr>                      Press Enter to execute the command.
<severitylevel>           Enter Logging Severity Level (emergency|0, alert|1,
                          critical|2, error|3, warning|4, notice|5, info|6,
                          debug|7).

   logging host 192.168.21.253 4 1 ?

<cr>                      Press Enter to execute the command.

   logging host 192.168.21.253 4 1

   exit

show logging hosts ?

<unit>                    Enter switch ID in the range of 1 to 8.

show logging hosts 1

Index     IP Address      Port     Status
-----  -----------------  ----  -------------
1      192.168.21.253     4     Active
```

1.7 Port Description

The Port Description feature lets you specify an alphanumeric interface identifier that can be used for SNMP network management.

1.7.1 CLI Example

Use the commands shown below for the Port Description feature.

1.7.1.1 Example #1: Enter a Description for a Port

This example specifies the name “Test” for port 0/10:

```
(Ethernet Fabric) #config
(Ethernet Fabric) (Config) #interface 0/10
(Ethernet Fabric) (Config) (Interface 0/10) #description Test
(Ethernet Fabric) (Config) (Interface 0/10) #exit
(Ethernet Fabric) (Config) #exit
```

1.7.1.2 Example #2: Show the Port Description

```
(Ethernet Fabric) #show port description 0/10

Interface.......0/10
ifIndex.........10
Description.....Test
MAC Address.....00:00:00:01:00:02
Bit Offset Val..10
```

1.8 Storm Control

A traffic storm is a condition that occurs when incoming packets flood the LAN, which creates performance degradation in the network. The FASTPATH Storm Control feature protects against this condition.

FASTPATH provides broadcast, multicast, and unicast storm recovery for individual interfaces or for all interfaces, depending on forwarding-plane silicon. If the silicon supports configuration for all interfaces, you will not be able to configure individual interfaces.

Unicast Storm Control protects against traffic whose MAC addresses are not known by the system.

For broadcast, multicast, and unicast storm control, if the rate of traffic ingressing on an interface increases beyond the configured threshold for that type, the traffic is dropped.

To configure storm control, you will enable the feature for all interfaces or for individual interfaces, and you will set the threshold (storm control level) beyond which the broadcast, multicast, or unicast traffic will be dropped.

Configuring a storm-control level also enables that form of storm-control. Disabling a storm-control level (using the “no” version of the command) sets the storm-control level back to default value and disables that form of storm-control. Using the “no” version of the “storm-control” command (not stating a “level”) disables that form of storm-control but maintains the configured “level” (to be active next time that form of storm-control is enabled).

1.8.1 CLI Example

1.8.1.1 Example #1: Set Broadcast Storm Control for All Interfaces

```
(Ethernet Fabric) #Config
(Ethernet Fabric) (Config) #storm-control broadcast ?

all                     Configure storm-control features for all ports.

(Ethernet Fabric) (Config) #storm-control broadcast all ?

<cr>                    Press Enter to execute the command.
level                   Configure storm-control thresholds.

(Ethernet Fabric) (Config) #storm-control broadcast all level ?

<rate>                  Enter the storm-control threshold as percent of port
                        speed.

(Ethernet Fabric) (Config) #storm-control broadcast all level 7
(Ethernet Fabric) (Config) #exit
```

1.8.1.2 Example #2: Set Multicast Storm Control for All Interfaces

```
(Ethernet Fabric) #Config
(Ethernet Fabric) (Config) #storm-control multicast all ?

<cr>                    Press Enter to execute the command.
level                   Configure storm-control thresholds.

(Ethernet Fabric) (Config) #storm-control multicast all level 8
(Ethernet Fabric) (Config) #exit
```

1.8.1.3 Example #3: Set Unicast Storm Control for All Interfaces

```
(Ethernet Fabric) #Config
(Ethernet Fabric) (Config) #storm-control unicast all
(Ethernet Fabric) (Config) #storm-control unicast all level 5
(Ethernet Fabric) (Config) #exit
```

1.9 Cable Test

The cable test feature enables you to determine the cable connection status on a selected port.

Note...

The cable test feature is supported only for copper cable. it is not supported for optical fiber cable.

In privileged exec mode, you enter `cablestatus` followed by the slot/port number of the port you want to test. One of the following statuses are returned:

- **Normal**: The cable is working correctly.
- **Open**: The cable is disconnected or there is a faulty connector.
- **Short**: There is an electrical short in the cable.
- **Cable Test Failed**: The cable status could not be determined. The cable may in fact be working.

The command may also return a cable length estimate if this feature is supported by the PHY for the current link speed. The length is displayed as a range between the shortest estimated length and the longest estimated length. Note that if the link is down and a cable is attached to a 10/100 Ethernet adapter, then the cable status may display as Open or Short because some Ethernet adapters leave unused wire pairs unterminated or grounded. Unknown is displayed if the cable length could not be determined.

If the port has an active link while the cable test is run, the link can go down for the duration of the test. The test may take several seconds to run.

1.9.1 CLI Example

```
(Ethernet Fabric) #cablestatus 0/34

Cable Status..................................... Normal
Cable Length..................................... 7m - 8m
```

2. Switching Configuration

This chapter provides configuration scenarios for the following features:

- Virtual LANs
- IGMP Snooping
- IGMP Proxy
- Link Aggregation/Port-channels
- Port Mirroring
- Port Security
- Link Layer Discovery Protocol
- Denial of Service Attack Protection
- DHCP Filtering
- Configuring Spanning Tree Protocol

2.1 Virtual LANs

Adding Virtual LAN (VLAN) support to a Layer 2 switch offers some of the benefits of both bridging and routing. Like a bridge, a VLAN switch forwards traffic based on the Layer 2 header, which is fast. Like a router, it partitions the network into logical segments, which provides better administration, security and management of multicast traffic.

A VLAN is a set of end stations and the switch ports that connect them. You can have many reasons for the logical division, for example, department or project membership. The only physical requirement is that the end station, and the port to which it is connected, both belong to the same VLAN.

Each VLAN in a network has an associated VLAN ID, which appears in the IEEE 802.1Q tag in the Layer 2 header of packets transmitted on a VLAN. An end station may omit the tag, or the VLAN portion of the tag, in which case the first switch port to receive the packet may either reject it or insert a tag using its default VLAN ID. A given port may handle traffic for more than one VLAN, but it can only support one default VLAN ID.

Two features let you define packet filters that the switch uses as the matching criteria to determine if a particular packet belongs to a particular VLAN.

- The IP-subnet Based VLAN feature lets you map IP addresses to VLANs by specifying a source IP address, network mask, and the desired VLAN ID.
- The MAC-based VLAN feature let packets originating from end stations become part of a VLAN according to source MAC address. To configure the feature, you specify a source MAC address and a VLAN ID.

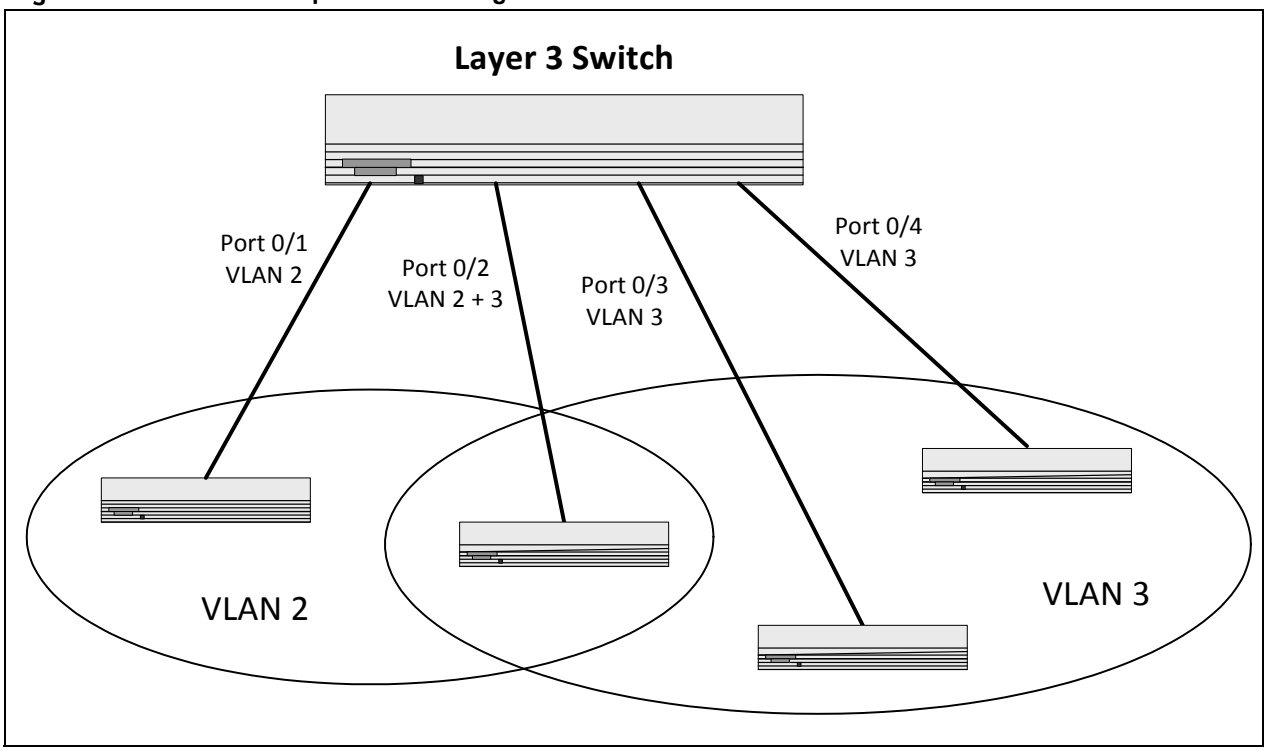
The Private Edge VLAN feature lets you set protection between ports located on the switch. This means that a protected port cannot forward traffic to another protected port on the same switch.

The feature does not provide protection between ports located on different switches.

2.1.1 VLAN Configuration Example

The diagram in this section shows a switch with four ports configured to handle the traffic for two VLANs. Port 0/2 handles traffic for both VLANs, while port 0/1 is a member of VLAN 2 only, and ports 0/3 and 0/4 are members of VLAN 3 only. The script following the diagram shows the commands you would use to configure the switch as shown in the diagram.

Figure 2-1: VLAN Example Network Diagram



2.1.2 CLI Examples

The following examples show how to create VLANs, assign ports to the VLANs, and assign a VLAN as the default VLAN to a port.

2.1.2.1 Example #1: Create Two VLANs

Use the following commands to create two VLANs and to assign the VLAN IDs while leaving the names blank.

```
(Ethernet Fabric)# vlan database
(Ethernet Fabric) (Vlan)# vlan 2
(Ethernet Fabric) (Vlan)# vlan 3
(Ethernet Fabric) (Vlan)# exit
```

2.1.2.2 Example #2: Assign Ports to VLAN2

This sequence shows how to assign ports to VLAN2, specify that frames will always be transmitted tagged from all member ports, and that untagged frames will be rejected on receipt.

```
(Ethernet Fabric) #Config
(Ethernet Fabric) (Config)# interface 0/1
(Ethernet Fabric) (Config) (interface 0/1)# vlan participation include 2
(Ethernet Fabric) (Config) (interface 0/1)# vlan acceptframe vlanonly
(Ethernet Fabric) (Config) (interface 0/1)# exit
```

```
(Ethernet Fabric) (Config)# interface 0/2
(Ethernet Fabric) (Config) (interface 0/2)# vlan participation include 2
(Ethernet Fabric) (Config) (interface 0/2)# vlan acceptframe vlanonly
(Ethernet Fabric) (Config) (interface 0/2)# exit
(Ethernet Fabric) (Config)# exit

(Ethernet Fabric)# Config
(Ethernet Fabric) (Config)# vlan port tagging all 2
(Ethernet Fabric) (Config)# exit
```

2.1.2.3 Example #3: Assign Ports to VLAN3

This example shows how to assign the ports that will belong to VLAN 3, and to specify that untagged frames will be accepted on port 0/4.

Note that port 0/2 belongs to both VLANs and that port 0/1 can never belong to VLAN 3.

```
(Ethernet Fabric) #Config
(Ethernet Fabric) (Config)# interface 0/2
(Ethernet Fabric) (Config) (interface 0/2)# vlan participation include 3
(Ethernet Fabric) (Config) (interface 0/2)# exit
(Ethernet Fabric) (Config)# interface 0/3
(Ethernet Fabric) (Config) (interface 0/3)# vlan participation include 3
(Ethernet Fabric) (Config)# exit
(Ethernet Fabric) (Config)# interface 0/4
(Ethernet Fabric) (Config) (interface 0/4)# vlan participation include 3
(Ethernet Fabric) (Config) (interface 0/4)# exit
(Ethernet Fabric) (Config)# exit

(Ethernet Fabric)# Config
(Ethernet Fabric) (Config)# interface 0/4
(Ethernet Fabric) (Config) (interface 0/4)# vlan acceptframe all
(Ethernet Fabric) (Config) (interface 0/4)# exit
(Ethernet Fabric) (Config)# exit
```

2.1.2.4 Example #4: Assign VLAN3 as the Default VLAN

This example shows how to assign VLAN 3 as the default VLAN for port 0/2.

```
(Ethernet Fabric) Config
(Ethernet Fabric) (Config)# interface 0/2
(Ethernet Fabric) (Interface 0/2)# vlan pvid 3
(Ethernet Fabric) (Interface 0/2)# exit
(Ethernet Fabric) (Config)# exit
```

2.1.2.5 Example #5: Assign IP Addresses to VLAN 2

```
(Ethernet Fabric)# vlan database
(Ethernet Fabric) (Vlan)# vlan association subnet 192.168.10.10 255.255.255.0 2
(Ethernet Fabric) (Vlan)# exit
(Ethernet Fabric)# show vlan association subnet

IP Address          IP Mask            VLAN ID
----------------    ----------------   -------

192.168.10.10       255.255.255.0      2
```

2.1.3 Private Edge VLANs

Use the Private Edge VLAN feature to prevent ports on the switch from forwarding traffic to each other even if they are on the same VLAN.

- Protected ports cannot forward traffic to other protected ports in the same group, even if they have the same VLAN membership. Protected ports can forward traffic to unprotected ports.
- Unprotected ports can forward traffic to both protected and unprotected ports.

You can also configure groups of protected ports, but unprotected ports are independent and cannot be added to a group. Each group's configuration consists of a name and a mask of ports. A port can belong to only one set of protected ports, but an unprotected port can be added to a group as a protected port.

The group name is configurable by the network administrator.

Use the **switchport protected** command to designate a port as protected. Use the **show switchport protected** command to display a listing of the protected ports.

2.1.4 CLI Example

2.1.4.1 Example #1: switchport protected

```
(Ethernet Fabric) #Config
(Ethernet Fabric) (config) #interface 0/1
(Ethernet Fabric) (config) (Interface 0/1) #switchport protected ?

<cr>          Press Enter to execute the command.

(Ethernet Fabric) (config) (Interface 0/1) #switchport protected 0
(Ethernet Fabric) (config) (Interface 0/1) #exit
```

2.1.4.2 Example #2: show switchport protected

```
(Ethernet Fabric) #show switchport protected
0/1
```

2.2 IGMP Snooping

This section describes the Internet Group Management Protocol (IGMP) Snooping feature. IGMP Snooping enables the switch to monitor IGMP transactions between hosts and routers. It can help conserve bandwidth by allowing the switch to forward IP multicast traffic only to connected hosts that request multicast traffic.

2.2.1 Overview

The IGMP feature:

- Uses Version 3 of IGMP
- Includes snooping, which can be enabled per VLAN

2.2.2 CLI Examples

The following are examples of the commands used in the IGMP Snooping feature.

2.2.2.1 Example #1: show igmpsnooping

```
(Ethernet Fabric) #show igmpsnooping ?

<cr>                Press Enter to execute the command.
<slot/port>         Enter interface in slot/port format.
mrouter             Display IGMP Snooping Multicast Router information.
<1-4093>            Display IGMP Snooping valid VLAN ID information.

(Ethernet Fabric) #show igmpsnooping

Admin Mode.............................Enable
Multicast Control Frame Count..........0
Interfaces Enabled for IGMP Snooping.....0/10
Vlans enabled for IGMP snooping.........20
```

2.2.2.2 Example #2: show ip igmp interface

```
(Ethernet Fabric) #show ip igmp interface ?

<slot/port>         Enter interface in slot/port format.
membership          Display interfaces subscribed to the multicast group.
stats               Display IGMP statistical information.

(Ethernet Fabric) #show ip igmp interface 0/10

Slot/Port.....................................0/10
IGMP Admin Mode...............................Enable
Interface Mode................................Disable
IGMP Version..................................3
Query Interval (secs).........................125
Query Max Response Time (1/10 of a second).....100
Robustness....................................2
Startup Query Interval (secs).................31
Startup Query Count...........................2
Last Member Query Interval (1/10 of a second)..10
Last Member Query Count.......................2
```

2.2.2.3 Example #3: show mac-address-table igmpsnooping

```
(Ethernet Fabric) #show mac-address-table igmpsnooping

Type                        Description               Interfaces
------------------------------------------------------------------------
00:01:01:00:5E:00:01:16     DynamicNetwork AssistFwd: 0/47
00:01:01:00:5E:00:01:18     DynamicNetwork AssistFwd: 0/47
00:01:01:00:5E:37:96:D0     DynamicNetwork AssistFwd: 0//47
00:01:01:00:5E:7F:FF:FA     DynamicNetwork AssistFwd: 0/47
00:01:01:00:5E:7F:FF:FE     DynamicNetwork AssistFwd: 0/47
```

2.2.2.4 Example #4: show ip igmp interface

```
(Ethernet Fabric) #show ip igmp interface 0/2

Slot/Port..................................... 0/2
IGMP Admin Mode............................... Disable
Interface Mode................................ Disable
IGMP Version.................................. 3
Query Interval (secs)......................... 125
Query Max Response Time (1/10 of a second)..... 100
Robustness.................................... 2
Startup Query Interval (secs) ................ 31
Startup Query Count........................... 2
Last Member Query Interval (1/10 of a second).. 10
Last Member Query Count....................... 2
```

2.2.2.5 Example #5: (Config) #ip igmp

```
(Ethernet Fabric) (Config) #ip igmp
```

2.2.2.6 Example #6: #show ip igmp

```
(Ethernet Fabric) #show ip igmp ?

<cr>                    Press Enter to execute the command.
groups                  Display the subscribed multicast groups.
interface               Display IGMP configuration information.
```

2.2.2.7 Example #7: (Interface 0/2) #ip igmp

```
(Ethernet Fabric) (config) (Interface 0/2) #ip igmp ?

<cr>                        Press Enter to execute the command.
last-member-query-count     Configure last member query count.
last-member-query-interval Configure last member query interval.
query-interval              Configure IGMP query interval.
query-max-response-time     Configure maximum response time.
robustness                  Configure IGMP router robustness.
startup-query-count         Configure startup query count.
startup-query-interval      Configure startup query interval.
version                     Configure IGMP or IGMP Proxy version.
```

2.3 IGMP Proxy

This section describes the Internet Group Management Protocol (IGMP) Proxy feature. This feature allows to setup forwarding of IGMP messages on a physical or port-channel interface. All received membership reports are forwarded at once, the leave reports are only sent if the last member has left the group. The command is similar to setting an 'mrouter', but multicast traffic will not be forwarded to the interface by default.

The forwarding of IGMP messages is done independent of any VLAN configuration. If the forwarding should be done with considering the VLAN configuration, the parameter "vlan-aware" must be specified. In this case the IGMP message is only forwarded if the forwarding port (egress) is member of the related (ingress) VLAN.

2.3.1 CLI examples

2.3.1.1 Example #1: setup without VLAN aware

```
(Ethernet Fabric)# configure
(Ethernet Fabric) (Config)# interface 0/1
(Ethernet Fabric) (Config) (interface 0/1)# set igmp proxy-report interfacemode
(Ethernet Fabric) (Config) (interface 0/1)# exit
(Ethernet Fabric) (Config)# exit
```

2.3.1.2 Example #2: delete

```
(Ethernet Fabric)# configure
(Ethernet Fabric) (Config)# interface 0/1
(Ethernet Fabric) (Config) (interface 0/1)# no set igmp proxy-report interfacemode
(Ethernet Fabric) (Config) (interface 0/1)# exit
(Ethernet Fabric) (Config)# exit
```

2.3.1.3 Example #3: setup with VLAN aware

```
(Ethernet Fabric)# configure
(Ethernet Fabric) (Config)# interface 0/1
(Ethernet Fabric) (Config) (interface 0/1)# set igmp proxy-report interfacemode vlan-
aware
(Ethernet Fabric) (Config) (interface 0/1)# exit
(Ethernet Fabric) (Config)# exit
```

or

```
(Ethernet Fabric)# configure
(Ethernet Fabric) (Config)# interface 0/1
(Ethernet Fabric) (Config) (interface 0/1)# set igmp proxy-report interfacemode
(Ethernet Fabric) (Config) (interface 0/1)# set igmp proxy-report interfacemode vlan-
aware
(Ethernet Fabric) (Config) (interface 0/1)# exit
(Ethernet Fabric) (Config)# exit
```

2.3.1.4 Example #4: delete

If using the 'no' command with parameter "vlan-aware" only the vlan aware flag is deleted but the proxy setting is kept. If using the 'no' command without parameter, the proxy setting is deleted.

```
(Ethernet Fabric)# configure
(Ethernet Fabric) (Config)# interface 0/1
(Ethernet Fabric) (Config) (interface 0/1)# no set igmp proxy-report interfacemode
(Ethernet Fabric) (Config) (interface 0/1)# exit
(Ethernet Fabric) (Config)# exit
```

delete VLAN awareness:

```
(Ethernet Fabric)# configure
(Ethernet Fabric) (Config)# interface 0/1
(Ethernet Fabric) (Config) (interface 0/1)# no set igmp proxy-report interfacemode vlan-
aware
(Ethernet Fabric) (Config) (interface 0/1)# exit
(Ethernet Fabric) (Config)# exit
```

2.4 Link Aggregation/Port-channels

This section shows how to use the Link Aggregation feature to configure port-channels via the Command Line Interface and the Graphical User Interface.

The Link Aggregation (LAG) feature allows the switch to treat multiple physical links between two end-points as a single logical link called a port-channel. All of the physical links in a given port-channel must operate in full-duplex mode at the same speed.

You can use the feature to directly connect two switches when the traffic between them requires high bandwidth and reliability, or to provide a higher bandwidth connection to a public network.

You can configure the port-channels as either dynamic or static. Dynamic configuration uses the IEEE 802.3ad standard, which provides for the periodic exchanges of LACPDUs. Static configuration is used when connecting the switch to an external switch that does not support the exchange of LACPDUs.

The feature offers the following benefits:

- Increased reliability and availability: If one of the physical links in the port-channel goes down, traffic is dynamically and transparently reassigned to one of the other physical links.
- Increased bandwidth: The aggregated physical links deliver higher bandwidth than each individual link.
- Incremental increase in bandwidth: A physical upgrade could produce a 10-times increase in bandwidth; LAG produces a two- or five-times increase, useful if only a small increase is needed.

Management functions treat a port-channel as if it were a single physical port.

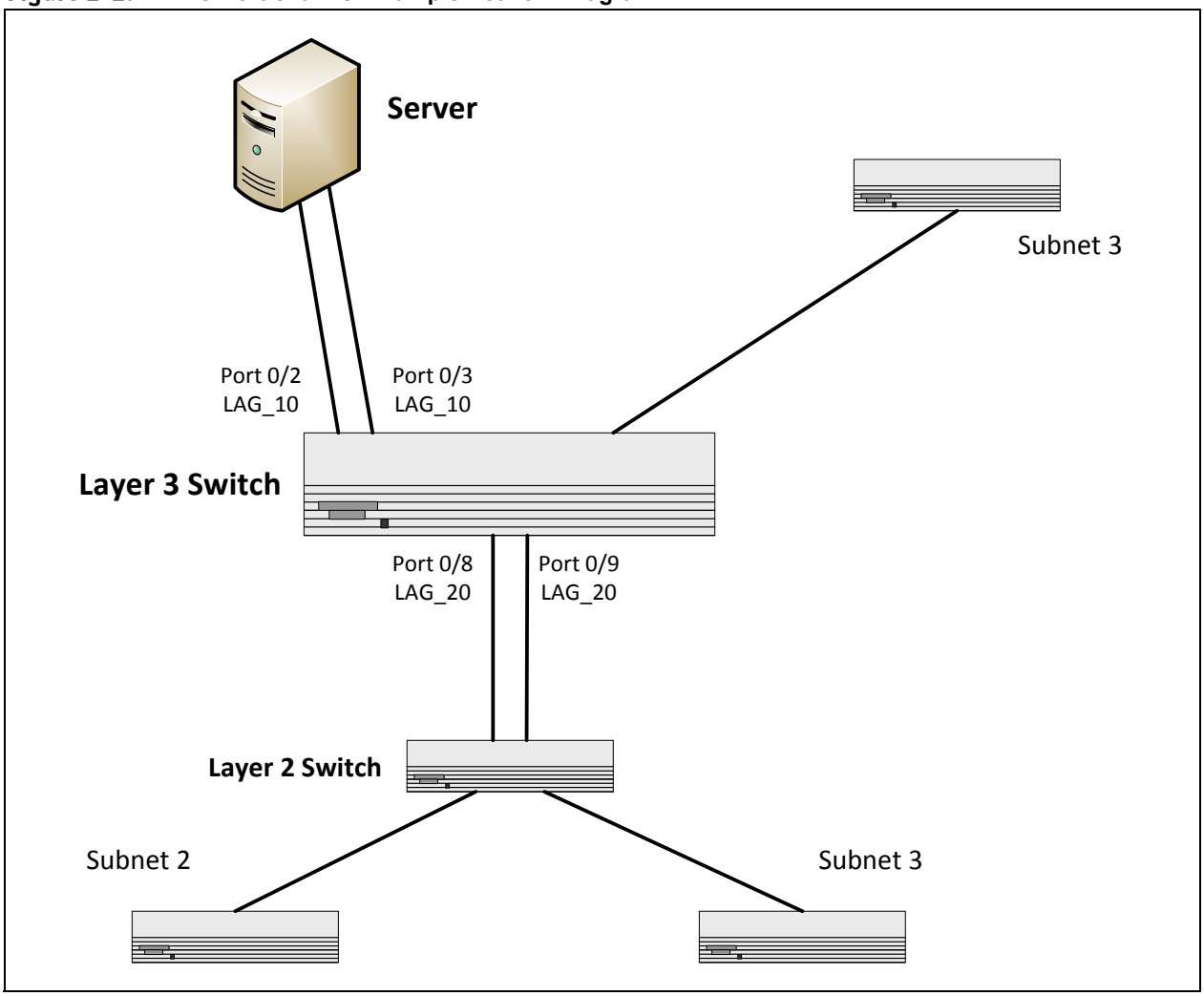
You can include a port-channel in a VLAN. You can configure more than one port-channel for a given switch.

2.4.1 CLI Example

The following shows an example of configuring the software to support Link Aggregation (LAG) to a server and to a Layer 3 switch.

The figure on the following page shows the example network.

Figure 2-2: LAG/Port-channel Example Network Diagram



2.4.1.1 Example 1: Create two port-channels:

Depending on the used FP version, port channels have to setup manually or are already predefined by default, so the following config commands may not be needed. Please check with "show port-channel all".

```
(Ethernet Fabric) #Config
(Ethernet Fabric) (Config) #port-channel lag_10
(Ethernet Fabric) (Config) #port-channel lag_20
(Ethernet Fabric) (Config) #exit
```

Use the **show port-channel all** command to show the logical interface ids you will use to identify the port-channels in subsequent commands. Assume that lag_10 is assigned id 1/1 and lag_20 is assigned id 1/2.

```
(Ethernet Fabric) #show port-channel all

              Port-                 Link
Log.          Channel          Adm. Trap  STP            Mbr       Port     Port
Intf           Name      Link  Mode Mode  Mode   Type    Ports     Speed    Active
------ -------------- ----- ---- ---- ------ -------- ------ ---------- ------
1/1    lag_10         Down  En.  En.  Dis.   Dynamic
1/2    lag_20         Down  En.  En.  Dis.   Dynamic
```

2.4.1.2 Example 2: Add the physical ports to the port-channels:

subsequent commands. Assume that lag_10 is assigned id 1/1 and lag_20 is assigned id 1/2.

```
(Ethernet Fabric) #Config
(Ethernet Fabric) (Config) #interface 0/2
(Ethernet Fabric) (Config) (interface 0/2)# addport 1/1
(Ethernet Fabric) (Config) (interface 0/2)# exit

(Ethernet Fabric) (Config) #interface 0/3
(Ethernet Fabric) (Config) (interface 0/3) #addport 1/1
(Ethernet Fabric) (Config) (interface 0/3) #exit

(Ethernet Fabric) (Config) #exit

(Ethernet Fabric) #Config
(Ethernet Fabric) (Config) #interface 0/8
(Ethernet Fabric) (Config) (interface 0/8) #addport 1/2
(Ethernet Fabric) (Config) (interface 0/8) #exit

(Ethernet Fabric) (Config) #interface 0/9
(Ethernet Fabric) (Config) (interface 0/9) #addport 1/2
(Ethernet Fabric) (Config) (interface 0/9) #exit

(Ethernet Fabric) (Config) #exit
```

2.4.1.3 Example 3: Enable both port-channels.

By default, the system enables link trap notification

```
(Ethernet Fabric) #Config
(Ethernet Fabric) (Config) #port-channel adminmode all
(Ethernet Fabric) (Config) #exit
```

At this point, the LAGs could be added to the default management VLAN.

2.5 Port Mirroring

This section describes the Port Mirroring feature, which can serve as a diagnostic tool, debugging tool, or means of fending off attacks.

2.5.1 Overview

Port mirroring selects network traffic from specific ports for analysis by a network analyzer, while allowing the same traffic to be switched to its destination. You can also configure how traffic is mirrored on a source port. Packets received on the source port, transmitted on a port, or both received and transmitted, can be mirrored to the destination port.

You can configure many switch ports as source ports and one switch port as a destination port.

Note...

The traffic on the destination port is restricted to a dedicated bandwidth, so in case the aggregated bandwidth of the source ports is higher then the one of the destination port, you may loos packets.

2.5.2 CLI Examples

The following are examples of the commands used in the Port Mirroring feature.

2.5.2.1 Example #1: Set up a Port Mirroring Session

The following command sequence enables port mirroring and specifies a source and destination ports.

```
(Ethernet Fabric) #Config
(Ethernet Fabric) (Config) #monitor session 1 mode
(Ethernet Fabric) (Config) #monitor session 1 source interface 0/7 ?

<cr>                    Press Enter to execute the command.
rx                      Monitor ingress packets only.
tx                      Monitor egress packets only.

(Ethernet Fabric) (Config) #monitor session 1 source interface 0/7
(Ethernet Fabric) (Config) #monitor session 1 destination interface 0/8
(Ethernet Fabric) (Config) #exit
```

2.5.2.2 Example #2: Show the Port Mirroring Session

```
(Ethernet Fabric) #show monitor session 1

Session ID   Admin Mode   Probe Port   Mirrored Port   Type
----------   ----------   ----------   -------------   -----
1            Enable       0/8          0/7             Rx,Tx
```

Monitor session ID “1” - “1” is a hardware limitation.

2.5.2.3 Example #3: Show the Status of All Ports

```
(Ethernet Fabric) #show port all

                 Admin    Physical            Physical Link     Link     LACP
Intf     Type    Mode     Mode                Status   Status   Trap     Mode
----     ----      ------   --------  --------  ------   ----     ----
0/1              Enable   Auto                Up       Enable   Enable
0/2              Enable   Auto                Down     Enable   Enable
0/3              Enable   Auto                Down     Enable   Enable
0/4              Enable   Auto                Down     Enable   Enable
0/5              Enable   Auto                Down     Enable   Enable
0/6              Enable   Auto                Down     Enable   Enable
0/7     Mirror   Enable   Auto                Down     Enable   Enable
0/8     Probe    Enable   Auto                Down     Enable   Enable
0/9              Enable   Auto                Down     Enable   Enable
0/10             Enable   Auto                Down     Enable   Enable
```

2.5.2.4 Example #4: Show the Status of the Source and Destination Ports

Use this command for a specific port. The output shows whether the port is the mirror or the probe port, what is enabled or disabled on the port, etc.

```
(Ethernet Fabric) #show port 0/7

                 Admin    Physical  Physical  Link     Link     LACP
Intf    Type     Mode     Mode      Status    Status   Trap     Mode
----    ----     ------   --------  --------  ------   ----     ----
0/7     Mirror   Enable   Auto                Down     Enable   Enable


(Ethernet Fabric) #show port 0/8

                 Admin    Physical  Physical  Link     Link     LACP
Intf    Type     Mode     Mode      Status    Status   Trap     Mode
----    ----     ------   --------  --------  ------   ----     ----
0/8     Probe    Enable   Auto                Down     Enable   Enable
```

2.6 Port Security

This section describes the Port Security feature.

2.6.1 Overview

Port Security:

- Allows for limiting the number of MAC addresses on a given port.
- Packets that have a matching MAC address (secure packets) are forwarded; all other packets (unsecure packets) are restricted.
- Enabled on a per port basis.
- When locked, only packets with allowable MAC address will be forwarded.
- Supports both dynamic and static.
- Implement two traffic filtering methods. These methods can be used concurrently.
    - Dynamic Locking: User specifies the maximum number of MAC addresses that can be learned on a port. The maximum number of MAC addresses is platform (product) dependent. After the limit is reached, additional MAC addresses are not learned. Only frames with an allowable source MAC address are forwarded.
    - Static Locking: User manually specifies a list of static MAC addresses for a port. Dynamically locked addresses can be converted to statically locked addresses.

2.6.2 Operation

Port Security:

- Helps secure network by preventing unknown devices from forwarding packets.
- When link goes down, all dynamically locked addresses are 'freed.'
- If a specific MAC address is to be set for a port, set the dynamic entries to 0, then only allow packets with a MAC address matching the MAC address in the static list.
- Dynamically locked MAC addresses are aged out if another packet with that address is not seen within the age-out time. The user can set the time-out value.
- Dynamically locked MAC addresses are eligible to be learned by another port.
- Static MAC addresses are not eligible for aging.
- Dynamically locked addresses can be converted to statically locked addresses.

2.6.3 CLI Examples

The following are examples of the commands used in the Port Security feature.

2.6.3.1 Example #1: show port security

```
(Ethernet Fabric) #show port-security ?

<cr>                    Press Enter to execute the command.
all                     Display port-security information for all
                        interfaces
<slot/port>             Display port security information for a
                        specific interface.
dynamic                 Display dynamically learned MAC addresses.
static                  Display statically locked MAC addresses.
violation               Display the source MAC address of the last packet that was
discarded
                        on a locked port.
```

2.6.3.2 Example #2: show port security on a specific interface

```
(Ethernet Fabric) #show port-security 0/10

         Admin    Dynamic     Static    Violation
 Intf    Mode      Limit      Limit     Trap Mode
------  -------  ----------  ---------  ----------
0/10   Disabled 600         20         Disabled
```

2.6.3.3 Example #3: (Config) port security

```
(Ethernet Fabric) (Config) #port-security ?

<cr>Press Enter to execute the command.

(Ethernet Fabric) (Config) #port-security
```

2.7 Link Layer Discovery Protocol

The Link Layer Discovery Protocol (LLDP) feature allows individual interfaces on the switch to advertise major capabilities and physical descriptions. Network managers can view this information and identify system topology and detect bad configurations on the LAN.

LLDP has separately configurable transmit and receive functions. Interfaces can transmit and receive LLDP information.

2.7.1 CLI Examples

2.7.1.1 Example #1: Set Global LLDP Parameters

Use the following sequence to specify switch-wide notification interval and timers for all LLDP interfaces.

```
(Ethernet Fabric) #Config
(Ethernet Fabric) (Config) #lldp ?

notification-interval    Configure minimum interval to send remote data
                         change notifications
timers                   Configure the LLDP global timer values.

(Ethernet Fabric) (Config) #lldp notification-interval ?

<interval-seconds>       Range <5 - 3600> seconds.

(Ethernet Fabric) (Config) #lldp notification-interval 1000
(Ethernet Fabric) (Config) #lldp timers ?

<cr>                     Press Enter to execute the command.
hold                     The interval multiplier to set local LLDP data TTL.
interval                 The interval in seconds to transmit local LLDP data.
reinit                   The delay before re-initialization.

(Ethernet Fabric) (Config) #lldp timers hold 8 reinit 5
(Ethernet Fabric) (Config) #exit
```

2.7.1.2 Example #2: Set Interface LLDP Parameters

The following commands configure interface 0/10 to transmit and receive LLDP information.

```
(Ethernet Fabric) #Config
(Ethernet Fabric) (Config) #interface 0/10
(Ethernet Fabric) (Config) (interface 0/10) #lldp ?

notification             Enable/Disable LLDP remote data change notifications.
receive                  Enable/Disable LLDP receive capability.
transmit                 Enable/Disable LLDP transmit capability.
transmit-mgmt            Include/Exclude LLDP management address TLV.
transmit-tlv             Include/Exclude LLDP optional TLV(s).

(Ethernet Fabric) (Config) (interface 0/10) #lldp receive
(Ethernet Fabric) (Config) (interface 0/10) #lldp transmit
(Ethernet Fabric) (Config) (interface 0/10) #lldp transmit-mgmt
(Ethernet Fabric) (Config) #exit
(Ethernet Fabric) #exit
```

2.7.1.3 Example #3: Show Global LLDP Parameters

```
(Ethernet Fabric) #show lldp

LLDP Global Configuration

Transmit Interval.............................. 30 seconds
Transmit Hold Multiplier....................... 8
Reinit Delay................................... 5 seconds
Notification Interval.......................... 1000 seconds
```

2.7.1.4 Example #4 Show Interface LLDP Parameters

```
(Ethernet Fabric) #show lldp interface 0/10

LLDP Interface Configuration

Interface  Link    Transmit  Receive   Notify    TLVs      Mgmt
---------  ------  --------  --------  --------  -------   ----
0/10       Down    Enabled   Enabled   Disabled            Y

TLV Codes: 0- Port Description,   1- System Name
           2- System Description, 3- System Capabilities
```

2.8 Denial of Service Attack Protection

This section describes the FASTPATH Denial of Service Protection feature.

2.8.1 Overview

Denial of Service:

- Spans two categories:
  - Protection of the host running FASTPATH
  - Protection of the network
- Protects against the exploitation of a number of vulnerabilities which would make the host or network unstable
- Compliant with Nessus. Broadcom tested FASTPATH with Nessus version 2.0.10. Nessus is a widely used vulnerability assessment tool.
- FASTPATH software provides a number of features that help a network administrator protect networks against DoS attacks.

2.8.2 CLI Examples

Enter from Global Config mode:

```
(Ethernet Fabric) (Config) #dos-control sipdip
(Ethernet Fabric) (Config) #dos-control firstfrag
(Ethernet Fabric) (Config) #dos-control tcpfrag
(Ethernet Fabric) (Config) #dos-control l4port
(Ethernet Fabric) (Config) #dos-control icmp
(Ethernet Fabric) #show dos-control
```

2.9 DHCP Filtering

This section describes the Dynamic Host Configuration Protocol (DHCP) Filtering feature.

2.9.1 Overview

DHCP filtering provides security by filtering untrusted DHCP messages. An untrusted message is a message that is received from outside the network or firewall, and that can cause traffic attacks within network.

You can use DHCP Filtering as a security measure against unauthorized DHCP servers. A known attack can occur when an unauthorized DHCP server responds to a client that is requesting an IP address. The unauthorized server can configure the gateway for the client to be equal to the IP address of the server. At that point, the client sends all of its IP traffic destined to other networks to the unauthorized machine, giving the attacker the possibility of filtering traffic for passwords or employing a 'man-in-the-middle' attack.

DHCP filtering works by allowing the administrator to configure each port as a trusted or untrusted port. The port that has the authorized DHCP server should be configured as a trusted port. Any DHCP responses received on a trusted port will be forwarded. All other ports should be configured as untrusted. Any DHCP (or BootP) responses received on the ingress side will be discarded.

2.9.2 Limitations

- Port Channels (LAGs): If an interface becomes a member of a LAG, DHCP filtering is no longer operationally enabled on the interface. Instead, the interface follows the configuration of the LAG port. End user configuration for the interface remains unchanged. When an interface is no longer a member of a LAG, the current end user configuration for that interface automatically becomes effective.
- Mirroring: If an interface becomes a probe port, DHCP filtering can no longer become operationally enabled on the interface. End user configuration for the interface remains unchanged. When an interface no longer acts as a probe port, the current end user configuration for that interface automatically becomes effective.
- Operation without DHCP Relay: On platforms in which the DHCP relay feature is not included, hardware support must be available for the DHCP Filtering feature to operate.
- DHCP Relay: When DHCP Filtering is administratively enabled, the DHCP relay function must check whether a port is trusted before a DHCP (or BootP) response is forwarded on the port. If the port is untrusted, the response is dropped. The forwarding of DHCP or BootP request is unaffected.

- If DHCP Filtering is administratively disabled, the operation of the DHCP relay function is unaffected.
- If hardware support is available for DHCP Filtering, DHCP Filtering may be enabled both routing and non-routing interfaces.
- If hardware support is unavailable, DHCP Filtering may be enabled only on routed interfaces and only on interfaces enabled for DHCP relay.

2.9.3 CLI Examples

The commands shown below show examples of configuring DHCP Filtering for the switch and for individual interfaces.

2.9.3.1 Example #1: Enable DHCP Filtering for the Switch

This example

```
(Ethernet Fabric) #config
(Ethernet Fabric) (Config) #ip dhcp filtering
(Ethernet Fabric) (Config) #exit
```

2.9.3.2 Example #2: Enable DHCP Filtering for an Interface

```
(Ethernet Fabric) #config
(Ethernet Fabric) (Config) #interface 0/11
(Ethernet Fabric) (Config) (interface 0/11) #ip dhcp filtering trust
(Ethernet Fabric) (Config) (interface 0/11) #exit
(Ethernet Fabric) (Config) #exit
```

2.9.3.3 Example #3: Show DHCP Filtering Configuration

```
(Ethernet Fabric) #show ip dhcp filtering

Switch DHCP Filtering is Enabled

Interface    Trusted
-----------  ----------
0/1          No
0/2          No
0/3          No
0/4          No
0/5          No
0/6          No
0/7          No
0/8          No
0/9          No
0/10         No
0/11         Yes
0/12         No
0/13         No
0/14         No
0/15         No
```

2.10 Configuring Spanning Tree Protocol

The Spanning Tree Protocol (STP) provides a tree topology for any arrangement of bridges. STP also provides one path between end stations on a network, eliminating loops. Spanning tree versions supported include Common STP, Multiple STP, and Rapid STP.

Classic STP provides a single path between end stations, avoiding and eliminating loops.

Multiple Spanning Tree Protocol (MSTP) supports multiple instances of Spanning Tree to efficiently channel VLAN traffic over different interfaces. Each instance of the Spanning Tree behaves in the manner specified in IEEE 802.1w, Rapid Spanning Tree (RSTP), with slight modifications in the working but not the end effect (chief among the effects, is the rapid transitioning of the port to 'Forwarding'). The difference between the RSTP and the traditional STP (IEEE 802.1D) is the ability to configure and recognize full duplex connectivity and ports which are connected to end stations, resulting in rapid transitioning of the port to 'Forwarding' state and the suppression of Topology Change Notification. These features are represented by the parameters 'pointtopoint' and 'edgeport'. MSTP is compatible to both RSTP and STP. It behaves appropriately to STP and RSTP bridges. A MSTP bridge can be configured to behave entirely as a RSTP bridge or a STP bridge.

Note...

For two bridges to be in the same region, the force version should be 802.1S and their configuration name, digest key, and revision level should match. For more information about regions and their effect on network topology, refer to the IEEE 802.1Q standard.

2.10.1 Configuring Spanning Tree Protocol

This example shows how to enable IEEE 802.1D Spanning Tree (MST) protocol on the switch.

Note...

Spanning tree protocols are disabled by default for the entire switch and for individual ports. When you enable spanning tree protocol operation on the switch, you must also enable it on individual ports for it to be fully activated. When spanning tree protocol operation is disabled, the switch does not forward BPDU messages.

The example commands assume that you begin in Privileged EXEC mode.

2.10.1.1 Example #1: Enable STP on dedicated port

```
(Ethernet Fabric) #config
(Ethernet Fabric) (Config) #spanning-tree
(Ethernet Fabric) (Config) #interface 0/5
(Ethernet Fabric) (Config) (interface 0/5) #spanning-tree port mode
(Ethernet Fabric) (Config) (interface 0/5) #exit
(Ethernet Fabric) (Config) #exit
(Ethernet Fabric) #show spanning-tree summary
  Spanning Tree Adminmode.......... Enabled
  ...
(Ethernet Fabric) #show spanning-tree interface 0/5
  Hello Time..................................... Not Configured
  Port Mode...................................... Enabled
  ...Spanning Tree Protocol Commands
```

2.10.1.2 Example #2: Enable stp on all ports

To enable spanning-tree for all ports, use the command spanning-tree port mode all.

```
(Ethernet Fabric) #config
(Ethernet Fabric) (Config) #spanning-tree
(Ethernet Fabric) (Config) #spanning-tree port mode all
(Ethernet Fabric) (Config) #exit
```

2.10.1.3 Example #3: Disable spanning tree

```
(Ethernet Fabric) #config
(Ethernet Fabric) (Config) #no spanning-tree
(Ethernet Fabric) (Config) #exit

(Ethernet Fabric) #show spanning-tree summary
   Spanning Tree Adminmode.......... Disabled
   ...
```

2.10.1.4 Example #4: Show spanning tree details

This example assumes that spanning-tree is enabled as done by Example #1: Enable spanning tree. Use the show spanning-tree command to show spanning tree settings for the common and internal spanning tree.

```
(Ethernet Fabric) #show spanning-tree
Bridge Priority.............................. 32768
Bridge Identifier............................ 80:00:00:00:50:19:58:91
Time Since Topology Change................... 0 day 0 hr 1 min 55 sec
Topology Change Count........................ 3
Topology Change in progress.................. FALSE
Designated Root.............................. 80:00:00:00:50:18:F4:25
Root Path Cost............................... 20000
Root Port Identifier......................... 80:07
Bridge Max Age............................... 20
Bridge Max Hops.............................. 20
Bridge Forwarding Delay...................... 15
Hello Time................................... 2
Bridge Hold Time............................. 3
CST Regional Root............................ 80:00:00:00:50:19:58:91
Regional Root Path Cost...................... 0

     Associated FIDs           Associated VLANs
     ---------------           ----------------
     1                         1
```

2.10.1.5 Example #5: Show spanning tree statistics

Use the “show spanning-tree interface” command to show the settings and parameters for a specific switch port within the common and internal spanning tree. This example specifies port 0/5.

```
(Ethernet Fabric) #show spanning-tree interface 0/5
   Hello Time................................... Not Configured
   Port Mode.................................... Enabled
   Port Up Time Since Counters Last Cleared..... 0 day 1 hr 48 min 26 sec
   STP BPDUs Transmitted........................ 0
   STP BPDUs Received........................... 0
   RSTP BPDUs Transmitted....................... 225
   RSTP BPDUs Received.......................... 3023
   MSTP BPDUs Transmitted....................... 0
   MSTP BPDUs Received.......................... 0
```

2.10.1.6 Example #6: Show spanning tree summary

Use the “show spanning-tree summary” command to show spanning tree protocol version, settings and parameters for the switch. Notice that the default spanning tree protocol version is 802.1w, Rapid Spanning Tree Protocol.

```
(Ethernet Fabric) #show spanning-tree summary
   Spanning Tree Adminmode.......... Enabled
   Spanning Tree Version............ IEEE 802.1w
   Configuration Name............... 00-00-50-19-58-91
   Configuration Revision Level..... 0
   Configuration Digest Key......... 0xac36177f50283cd4b83821d8ab26de62
   Configuration Format Selector..... 0
   No MST instances to display.
```

2.10.1.7 Example #7: Configure bridge priority

This procedure shows how to configure the Spanning Tree Bridge Priority. In the spanning tree protocol, the root bridge is “elected” through a process where each port broadcasts information about itself (using bridge protocol data units or BPDUs). Each is identified by a priority and its MAC address. If no bridge priorities are explicitly specified, the root bridge is the one with the lowest MAC address burned in at the factory. If specified, the priority value is treated as more significant than the MAC address value. Therefore, specifying priority values allows you to directly control which switch is elected as root bridge, as well as to indirectly control which ports the other switches will put into the disabled state to prune a multi-connected (mesh) topology into the desired spanning tree.Spanning Tree Protocol Commands

In this example, the priority is 30000 and the MST instance is 0.

```
(Ethernet Fabric) #config
(Ethernet Fabric) (Config) #spanning-tree mst priority 0 30000

   Priority specified was converted to 28672 (according to IEEE 802.1s) and stored
   successfully.

(Ethernet Fabric) (Config) #exit

(Ethernet Fabric) #show spanning-tree
   The conversion occurred because the bridge priority must be a multiple of 4096.
```

2.10.1.8 Example #8: Configure port priority

This example displays details of port 0/2 and then configures its priority to 120. The MST instance 0 is the common and internal spanning tree.

```
(Ethernet Fabric) #show spanning-tree mst port detailed 0 0/2

Port Identifier.............................. 80:02
Port Priority................................ 128
Port Forwarding State........................ Forwarding
.
.
.
CST Path Cost................................ 0

(Ethernet Fabric) #config
(Ethernet Fabric) (Config) #interface 0/2
(Ethernet Fabric) (Config) (Interface 0/2) #spanning-tree mst 0 port-priority 120
```

```
Priority specified was converted to 112 (according to IEEE 802.1s) and store
successfully.

(Ethernet Fabric) (Config) (Interface 0/2) #exit
(Ethernet Fabric) (Config) #exit

(Ethernet Fabric) #show spanning-tree mst port detailed 0 0/2

Port Identifier................................ 70:02
Port Priority.................................. 112

The conversion occurred because the port priority must be a multiple of 16.
```

2.10.1.9 Example #9: Change to another spanning tree protocol version

You can switch from the default spanning tree protocol version, RSTP, to either STP or MSTP. RSTP offers the advantage over STP of using acknowledgements to more quickly reach a stable topology. MSTP offers the advantage over RSTP of supporting a different spanning tree for each group of VLANs, thus making use of bandwidth on links that might otherwise be unused.

The protocol versions are identified by the original names of the IEEE standards that specified them:

- 802.1d for STP
- 802.1w for RSTP
- 802.1s for MSTP

This example displays the current spanning tree settings and then changes the spanning tree protocol version from RSTP to MSTP.

```
(Ethernet Fabric) #show spanning-tree summary

Spanning Tree Adminmode............ Enabled
Spanning Tree Version.............. IEEE 802.1w
...

(Ethernet Fabric) #config
(Ethernet Fabric) (Config) #spanning-tree forceversion 802.1s
(Ethernet Fabric) (Config) #exit

(Ethernet Fabric) #show spanning-tree summary

Spanning Tree Adminmode............ Enabled
Spanning Tree Version.............. IEEE 802.1s
```

2.10.1.10 Example #10: Configuring Multiple Spanning Tree Protocol

This example shows how to enable IEEE 802.1s Multiple Spanning Tree (MST) protocol on the switch and all of the ports and to set the bridge priority.

To make multiple switches be part of the same MSTP region, make sure the Force Protocol Version setting for all switches is IEEE 802.1s. Also, make sure the configuration name, digest key, and revision level are the same for all switches in the region.

Note...

The digest key is generated based on the association of VLANs to different instances. To ensure the digest key is same, the mapping of VLAN to instance must be the same on each switch in the region. For example, if VLAN 10 is associated with instance 10 on one switch, you must associate VLAN 10 and instance 10 on the other switches.

Create VLAN 10 and VLAN 20.

```
(Ethernet Fabric) #vlan database
(Ethernet Fabric) (Vlan) #vlan 10
(Ethernet Fabric) (Vlan) #vlan 20
(Ethernet Fabric) (Vlan) #exit
```

Enable spanning tree Globally

```
(Ethernet Fabric) #config
(Ethernet Fabric) (Config) #spanning-tree
```

Create MST instances 10 and 20.

```
(Ethernet Fabric) (Config) #spanning-tree mst instance 10
(Ethernet Fabric) (Config) #spanning-tree mst instance 20
```

Associate MST instance 10 to VLAN 10 and MST instance 20 to VLAN 20

```
(Ethernet Fabric) (Config) #spanning-tree mst vlan 10 10
(Ethernet Fabric) (Config) #spanning tree mst vlan 20 20
```

Change the name so that all the bridges that want to be part of the same region can form the region.

```
(Ethernet Fabric) (Config) #spanning-tree configuration name broadcom
```

Make the MST ID 10 bridge the root bridge by lowering the priority.

```
(Ethernet Fabric) (Config) #spanning-tree mst priority 10 16384
```

Change the priority of MST ID 20 to ensure the other bridge is the root bridge.

```
(Ethernet Fabric) (Config) #spanning-tree mst priority 20 61440
```

Enable STP on interface 0/1

```
(Ethernet Fabric) (Config) #interface 0/1
(Ethernet Fabric) (Config) (Interface 0/1) #spanning-tree port mode
(Ethernet Fabric) (Config) (Interface 0/1) #exit
```

Enable STP on interface 0/2

```
(Ethernet Fabric) (Config) #interface 0/2
(Ethernet Fabric) (Config) (Interface 0/2) #spanning-tree port mode
```

On the non-root bridge change the priority to force port 0/2 to be the root port.

```
(Ethernet Fabric) (Config) (Interface 0/2) #spanning-tree mst 20 port-priority 64
(Ethernet Fabric) (Config) (Interface 0/2) #exit
```

3. Routing Configuration

This chapter describes configuration scenarios and instructions for the following routing features:

- Port Routing
- VLAN Routing
- Virtual Router Redundancy Protocol
- Proxy Address Resolution Protocol (ARP)
- OSPF
- Routing Information Protocol
- Route Preferences
- Loopback Interfaces

3.1 Port Routing

The first networks were small enough for the end stations to communicate directly. As networks grew, Layer 2 bridging was used to segregate traffic, a technology that worked well for unicast traffic, but had problems coping with large quantities of multicast packets. The next major development was routing, where packets were examined and redirected at Layer 3. End stations needed to know how to reach their nearest router, and the routers had to understand the network topology so that they could forward traffic. Although bridges tended to be faster than routers, using routers allowed the network to be partitioned into logical subnetworks, which restricted multicast traffic and also facilitated the development of security mechanisms.

An end station specifies the destination station's Layer 3 address in the packet's IP header, but sends the packet to the MAC address of a router. When the Layer 3 router receives the packet, at minimum it does the following:

- Looks up the Layer 3 address in its address table to determine the outbound port
- Updates the Layer 3 header
- Recreates the Layer 2 header

The router's IP address is often statically configured in the end station, although FASTPATH software supports protocols such as DHCP that allow the address to be assigned dynamically. Likewise, you may assign some of the entries in the routing tables used by the router statically, but protocols such as RIP and OSPF allow the tables to be created and updated dynamically as the network configuration changes.

3.1.1 Port Routing Configuration

FASTPATH software always supports Layer 2 bridging, but Layer 3 routing must be explicitly enabled, first for the FASTPATH as a whole, and then for each port which is to participate in the routed network.

The configuration commands used in the example in this section enable IP routing on ports 0/2, 0/3, and 0/5. The router ID is set to the FASTPATH switch's management IP address, or to that of any active router interface if the management address is not configured.

After you've issued the routing configuration commands, the following functions are active:

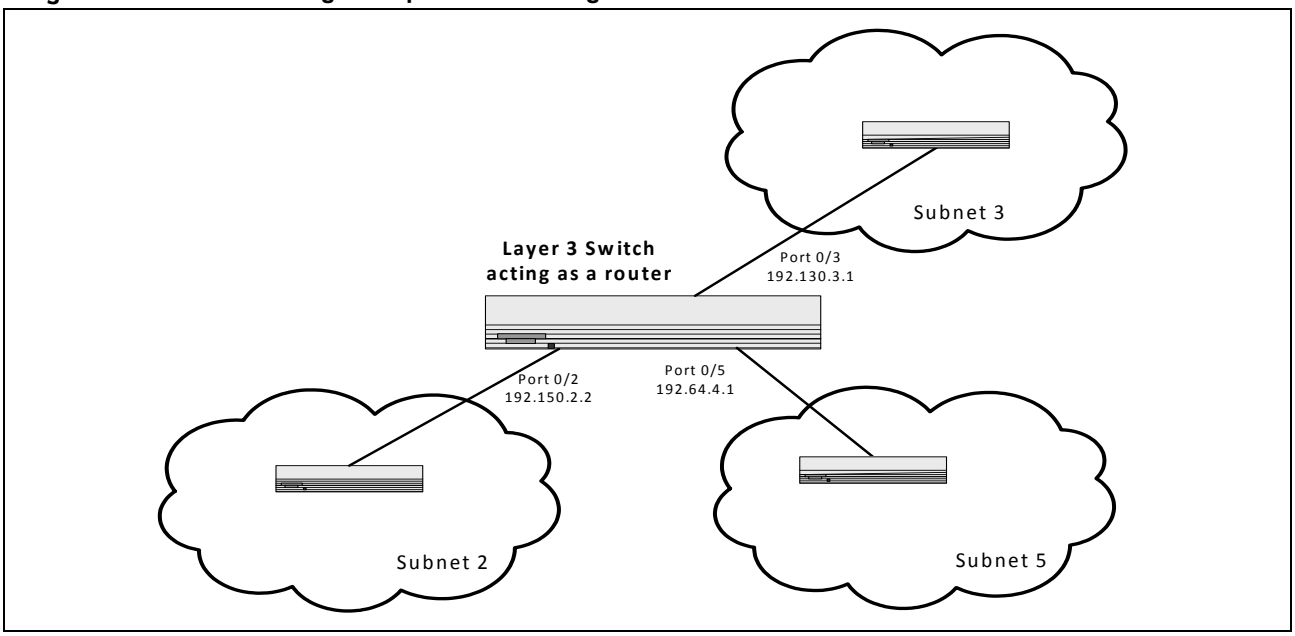
- IP Forwarding: responsible for forwarding received IP packets.
- ARP: responsible for maintaining the ARP Table used to correlate IP and MAC addresses. The table contains both static entries and entries dynamically updated based on information in received ARP frames.
- Routing Table Object: responsible for maintaining the common routing table used by all registered routing protocols.

You can then activate RIP or OSPF, used by routers to exchange route information, on top of IP Routing. RIP is most often used in smaller networks, while OSPF is most often used for larger and more complex topologies.

3.1.1.1 CLI Examples

The diagram in this section shows a Layer 3 switch configured for port routing. It connects three different subnets, each connected to a different port. The script shows the commands you would use to configure a FASTPATH switch to provide the port routing support shown in the diagram.

Figure 3-1: Port Routing Example Network Diagram



3.1.1.1.1 Example 1. Enabling Routing for the Switch

Use the following command to enable routing for the switch. Execution of the command enables IP forwarding by default.

```
(Ethernet Fabric) #config
(Ethernet Fabric) (Config) #ip routing
(Ethernet Fabric) (Config) #exit
```

3.1.1.1.2 Example 2. Enabling Routing for Ports on the Switch

Use the following commands to enable routing for ports on the switch. The default link-level encapsulation format is Ethernet. Configure the IP addresses and subnet masks for the ports. Network directed broadcast frames are dropped and the maximum transmission unit (MTU) size is 1500 bytes.

Please note, setting upPort based routing is not possible on all Kontron products, e.g. CP6923 does not provide this feature, but Routing must be configured based on VLANs.

```
(Ethernet Fabric) #config
(Ethernet Fabric) (Config) #interface 0/2
(Ethernet Fabric) (Config) (Interface 0/2) #routing
(Ethernet Fabric) (Config) (Interface 0/2) #ip address 192.150.2.2 255.255.255.0
(Ethernet Fabric) (Config) (Interface 0/2) #exit
(Ethernet Fabric) (Config) #exit

(Ethernet Fabric) (Config) #config
(Ethernet Fabric) (Config) #interface 0/3
(Ethernet Fabric) (Config) (Interface 0/2) #routing
(Ethernet Fabric) (Config) (Interface 0/2) #ip address 192.130.3.1 255.255.255.0
(Ethernet Fabric) (Config) (Interface 0/2) #exit
(Ethernet Fabric) (Config) #exit

(Ethernet Fabric) (Config) #config
(Ethernet Fabric) (Config) #interface 0/5
(Ethernet Fabric) (Config) (Interface 0/2) #routing
(Ethernet Fabric) (Config) (Interface 0/2) #ip address 192.64.4.1 255.255.255.0
(Ethernet Fabric) (Config) (Interface 0/2) #exit
(Ethernet Fabric) (Config) #exit
```

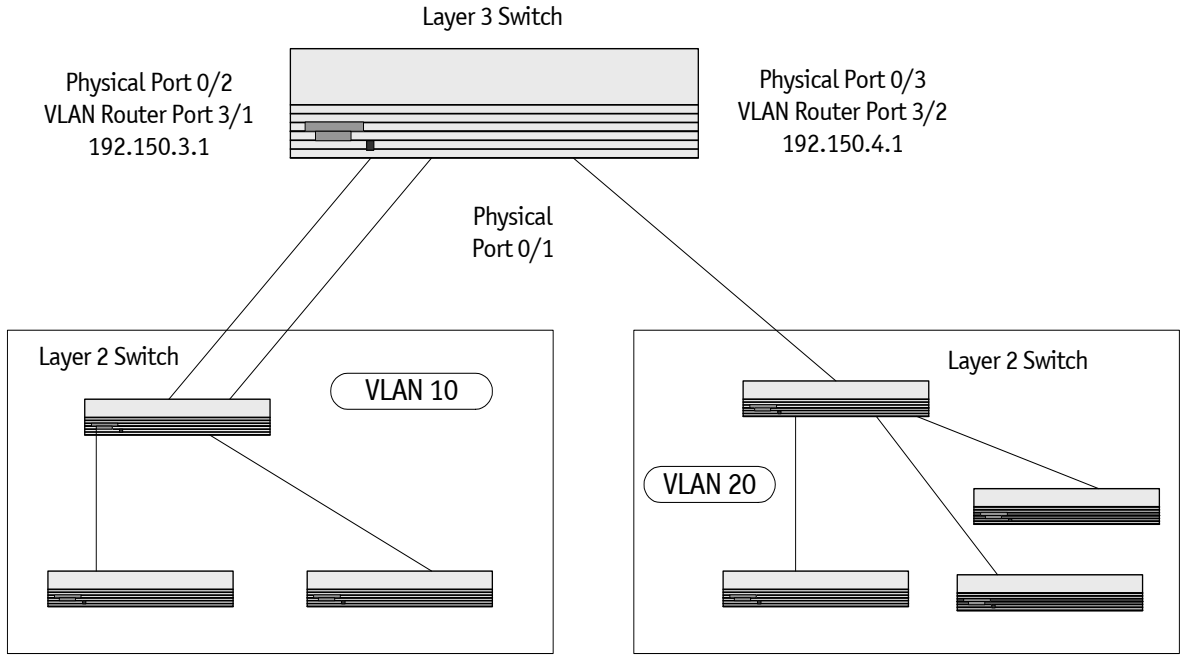
3.2 VLAN Routing

This section provides an example of how to configure FASTPATH software to support VLAN routing. The configuration of the VLAN router port is similar to that of a physical port. The main difference is that, after the VLAN has been created, you must use the **show ip vlan** command to determine the VLAN's interface ID so that you can use it in the router configuration commands.

3.2.1 CLI Examples

The diagram in this section shows a Layer 3 switch configured for port routing. It connects two VLANs, with two ports participating in one VLAN, and one port in the other. The script shows the commands you would use to configure FASTPATH software to provide the VLAN routing support shown in the following diagram.

Figure 3-2: VLAN Routing Example Network Diagram



3.2.1.1 Example 1: Create Two VLANs

The following code sequence shows an example of creating two VLANs with egress frame tagging enabled.

```
(Ethernet Fabric) #vlan database
(Ethernet Fabric) (Vlan) #vlan 10
(Ethernet Fabric) (Vlan) #vlan 20
(Ethernet Fabric) (Vlan) #exit

(Ethernet Fabric) #config
(Ethernet Fabric) (Config) #interface 0/1
(Ethernet Fabric) (Config) (Interface 0/1) #vlan participation include 10
(Ethernet Fabric) (Config) #exit
(Ethernet Fabric) (Config) #interface 0/2
(Ethernet Fabric) (Config) (Interface 0/1) #vlan participation include 10
(Ethernet Fabric) (Config) (Interface 0/1) #exit
(Ethernet Fabric) (Config) #interface 0/3
(Ethernet Fabric) (Config) (Interface 0/3) #vlan participation include 20
(Ethernet Fabric) (Config) (Interface 0/3) #exit
(Ethernet Fabric) (Config) #exit
(Ethernet Fabric) #config
(Ethernet Fabric) (Config) #vlan port tagging all 10
(Ethernet Fabric) (Config) #vlan port tagging all 20
(Ethernet Fabric) (Config) #exit
```

Next, specify the VLAN ID assigned to untagged frames received on the ports.

```
(Ethernet Fabric) #config
(Ethernet Fabric) (Config) #interface 0/1
(Ethernet Fabric) (Config) (Interface 0/1) #vlan pvid 10
(Ethernet Fabric) (Config) (Interface 0/1) #exit
(Ethernet Fabric) (Config) #interface 0/2
(Ethernet Fabric) (Config) (Interface 0/2) #vlan pvid 10
(Ethernet Fabric) (Config) (Interface 0/2) #exit
(Ethernet Fabric) (Config) #interface 0/3
(Ethernet Fabric) (Config) (Interface 0/3) #vlan pvid 20
(Ethernet Fabric) (Config) (Interface 0/3) #xit
(Ethernet Fabric) (Config) #exit
```

3.2.1.2 Example 2: Set Up VLAN Routing for the VLANs and the Switch.

The following code sequence shows how to enable routing for the VLANs:

```
(Ethernet Fabric) #vlan database
(Ethernet Fabric) (Vlan) #vlan routing 10
(Ethernet Fabric) (Vlan) #vlan routing 20
(Ethernet Fabric) (Vlan) #exit

(Ethernet Fabric) #show ip vlan
```

This returns the logical interface IDs that will be used instead of slot/port in subsequent routing commands. Assume that VLAN 10 is assigned ID 2/1 and VLAN 20 is assigned ID 2/2.

Enable routing for the switch:

```
(Ethernet Fabric) (Config) #config
(Ethernet Fabric) (Config) #ip routing
(Ethernet Fabric) (Config) #exit
```

The next sequence shows an example of configuring the IP addresses and subnet masks for the virtual router ports.

```
(Ethernet Fabric) (Config) #config
(Ethernet Fabric) (Config) #interface 2/1
(Ethernet Fabric) (Config) (Interface 2/1) #ip address 192.150.3.1 255.255.255.0
(Ethernet Fabric) (Config) (Interface 2/1) #exit
(Ethernet Fabric) (Config) #interface 2/2
(Ethernet Fabric) (Config) (Interface 2/2) #ip address 192.150.4.1 255.255.255.0
(Ethernet Fabric) (Config) (Interface 2/2) #exit
(Ethernet Fabric) (Config) #exit
```

3.2.2 VLAN Routing RIP Configuration

Routing Information Protocol (RIP) is one of the protocols which may be used by routers to exchange network topology information. It is characterized as an “interior” gateway protocol, and is typically used in small to medium-sized networks.

A router running RIP sends the contents of its routing table to each of its adjacent routers every 30 seconds. When a route is removed from the routing table it is flagged as unusable by the receiving routers after 180 seconds, and removed from their tables after an additional 120 seconds.

There are two versions of RIP:

- RIP-1 defined in RFC 1058
    - Routes are specified by IP destination network and hop count
    - The routing table is broadcast to all stations on the attached network
- RIP-2 defined in RFC 1723
    - Route specification is extended to include subnet mask and gateway
    - The routing table is sent to a multicast address, reducing network traffic
    - An authentication method is used for security

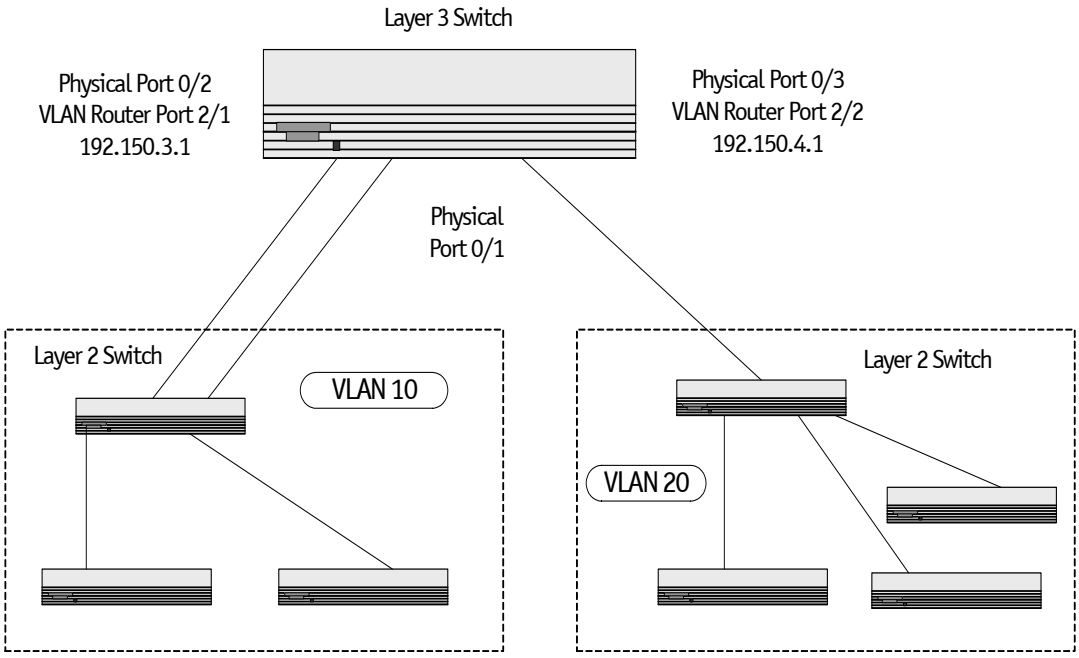
The FASTPATH software supports both versions of RIP. You may configure a given port:

- To receive packets in either or both formats
- To transmit packets formatted for RIP-1 or RIP-2 or to send RIP-2 packets to the RIP-1 broadcast address
- To prevent any RIP packets from being received
- To prevent any RIP packets from being transmitted

3.2.2.1 CLI Examples

This example adds support for RIP-2 to the configuration created in the base VLAN routing example.

Figure 3-3: RIP for VLAN Routing Example Network Diagram



3.2.2.1.1 Example 3: Configuring VLAN Routing with RIP Support

The following sequence creates the VLANs and enables VLAN routing.

```
(Ethernet Fabric) #vlan database
(Ethernet Fabric) (Vlan) #vlan 10
(Ethernet Fabric) (Vlan) #vlan 20
(Ethernet Fabric) (Vlan) #exit

(Ethernet Fabric) #config
(Ethernet Fabric) (Config) #interface 0/2
(Ethernet Fabric) (Config) (Interface 0/2) #vlan participation include 10
(Ethernet Fabric) (Config) (Interface 0/2) #exit
(Ethernet Fabric) (Config) #interface 0/3
(Ethernet Fabric) (Config) (Interface 0/3) #vlan participation include 20
(Ethernet Fabric) (Config) (Interface 0/3) #exit
(Ethernet Fabric) (Config) #exit

(Ethernet Fabric) #config
(Ethernet Fabric) (Config) #vlan port tagging all 10
(Ethernet Fabric) (Config) #vlan port tagging all 20
(Ethernet Fabric) (Config) #exit
```

```
(Ethernet Fabric) #config
(Ethernet Fabric) (Config) #interface 0/2
(Ethernet Fabric) (Config) (Interface 0/2) #vlan pvid 10
(Ethernet Fabric) (Config) (Interface 0/2) #exit
(Ethernet Fabric) (Config) #interface 0/3
(Ethernet Fabric) (Config) (Interface 0/3) #vlan pvid 20
(Ethernet Fabric) (Config) (Interface 0/3) #exit
(Ethernet Fabric) (Config) #exit

(Ethernet Fabric) #vlan database
(Ethernet Fabric) (Vlan) #vlan routing 10
(Ethernet Fabric) (Vlan) #vlan routing 20
(Ethernet Fabric) (Vlan) #exit

(Ethernet Fabric) #show ip vlan

(Ethernet Fabric) #config
(Ethernet Fabric) (Config) #ip routing
(Ethernet Fabric) (Config) #exit

(Ethernet Fabric) #config
(Ethernet Fabric) (Config) #interface 2/1
(Ethernet Fabric) (Config) (Interface 2/1) #ip address 192.150.3.1 255.255.255.0
(Ethernet Fabric) (Config) (Interface 2/1) #exit
(Ethernet Fabric) (Config) #interface 2/2
(Ethernet Fabric) (Config) (Interface 2/2) #ip address 192.150.4.1 255.255.255.0
(Ethernet Fabric) (Config) (Interface 2/2) #exit
(Ethernet Fabric) (Config) #exit
```

3.2.2.1.2 Example 4: Enable RIP for the Switch

This step enables RIP for the switch. The route preference will default to 120.

```
(Ethernet Fabric) #config
(Ethernet Fabric) (Config) #router rip
(Ethernet Fabric) (Router-config) #enable
(Ethernet Fabric) (Router-config) #exit
(Ethernet Fabric) (Config) #exit
```

The next sequence configures the IP address and subnet mask for a non-virtual router port.

```
(Ethernet Fabric) #config
(Ethernet Fabric) (Config) #interface 0/5
(Ethernet Fabric) (Config) (Interface 0/5) #ip address 192.150.5.1 255.255.255.0
(Ethernet Fabric) (Config) (Interface 0/5) #exit
(Ethernet Fabric) (Config) #exit
```

This last step enables RIP for the VLAN router ports. Authentication will default to none, and no default route entry will be created.

```
(Ethernet Fabric) #config
(Ethernet Fabric) (Config) #interface 2/1
(Ethernet Fabric) (Config) (Interface 2/1) #ip rip
(Ethernet Fabric) (Config) (Interface 2/1) #exit
(Ethernet Fabric) (Config) #interface 2/2
(Ethernet Fabric) (Config) (Interface 2/2) #ip rip
(Ethernet Fabric) (Config) (Interface 2/2) #exit
(Ethernet Fabric) (Config) #exit
```

3.2.3 VLAN Routing OSPF Configuration

For a overview of the OSPF protocol, see section „OSPF“.

3.2.3.1 CLI Example

This example adds support for OSPF to the configuration created in the base VLAN routing example. The script shows the commands you would use to configure the FASTPATH software as an inter-area router. Refer to Figure 3-3 .

3.2.3.1.1 Example 5: OSPF on FASTPATH as an Inter-area Router

Create the VLANs and enable VLAN routing.

```
(Ethernet Fabric) #vlan database
(Ethernet Fabric) (Vlan) #vlan 10
(Ethernet Fabric) (Vlan) #vlan 20
(Ethernet Fabric) (Vlan) #exit

(Ethernet Fabric) #config
(Ethernet Fabric) (Config) #interface 0/2
(Ethernet Fabric) (Config) (Interface 0/2) #vlan participation include 10
(Ethernet Fabric) (Config) (Interface 0/2) #exit
(Ethernet Fabric) (Config) #interface 0/3
(Ethernet Fabric) (Config) (Interface 0/3) #vlan participation include 20
(Ethernet Fabric) (Config) (Interface 0/3) #exit
(Ethernet Fabric) (Config) #exit

(Ethernet Fabric) #config
(Ethernet Fabric) (Config) #vlan port tagging all 10
(Ethernet Fabric) (Config) #vlan port tagging all 20
(Ethernet Fabric) (Config) #exit

(Ethernet Fabric) #config
(Ethernet Fabric) (Config) #interface 0/2
(Ethernet Fabric) (Config) (Interface 0/2) #vlan pvid 10
(Ethernet Fabric) (Config) (Interface 0/2) #exit
(Ethernet Fabric) (Config) #interface 0/3
(Ethernet Fabric) (Config) (Interface 0/3) #vlan pvid 20
(Ethernet Fabric) (Config) (Interface 0/3) #exit
(Ethernet Fabric) (Config) #exit

(Ethernet Fabric) #vlan database
(Ethernet Fabric) (Vlan) #vlan routing 10
(Ethernet Fabric) (Vlan) #vlan routing 20
(Ethernet Fabric) (Vlan) #exit

(Ethernet Fabric) #show ip vlan

(Ethernet Fabric) #config
(Ethernet Fabric) (Config) #ip routing
(Ethernet Fabric) (Config) #exit

(Ethernet Fabric) #config
(Ethernet Fabric) (Config) #interface 2/1
(Ethernet Fabric) (Config) (Interface 2/1) #ip address 192.150.3.1 255.255.255.0
(Ethernet Fabric) (Config) (Interface 2/1) #exit
(Ethernet Fabric) (Config) #interface 2/2
(Ethernet Fabric) (Config) (Interface 2/2) #ip address 192.150.4.1 255.255.255.0
(Ethernet Fabric) (Config) (Interface 2/2) #exit
(Ethernet Fabric) (Config) #exit
```

3.2.3.1.2 Example 6: Specify the Router ID and Enable OSPF for the Switch

```
(Ethernet Fabric) #config
(Ethernet Fabric) (Config) #router ospf
(Ethernet Fabric) (Config-router) #network 192.150.3.0 0.0.0.255 area 2
(Ethernet Fabric) (Config-router) #network 192.150.4.0 0.0.0.255 area 3
(Ethernet Fabric) (Config-router) #router-id 192.150.9.9
(Ethernet Fabric) (Config-router) #enable
(Ethernet Fabric) (Config-router) #exit
(Ethernet Fabric) (Config) #exit
```

Set the OSPF priority and cost for the VLAN and physical router ports:

```
(Ethernet Fabric) #config
(Ethernet Fabric) (Config) #interface 2/1
(Ethernet Fabric) (Config) (Interface 2/1) #ip ospf priority 128
(Ethernet Fabric) (Config) (Interface 2/1) #ip ospf cost 32
(Ethernet Fabric) (Config) (Interface 2/1) #exit
(Ethernet Fabric) (Config) #interface 2/2
(Ethernet Fabric) (Config) (Interface 2/2) #ip ospf priority 255
(Ethernet Fabric) (Config) (Interface 2/2) #ip ospf cost 64
(Ethernet Fabric) (Config) (Interface 2/2) #exit
(Ethernet Fabric) (Config) #exit
```

3.3 Virtual Router Redundancy Protocol

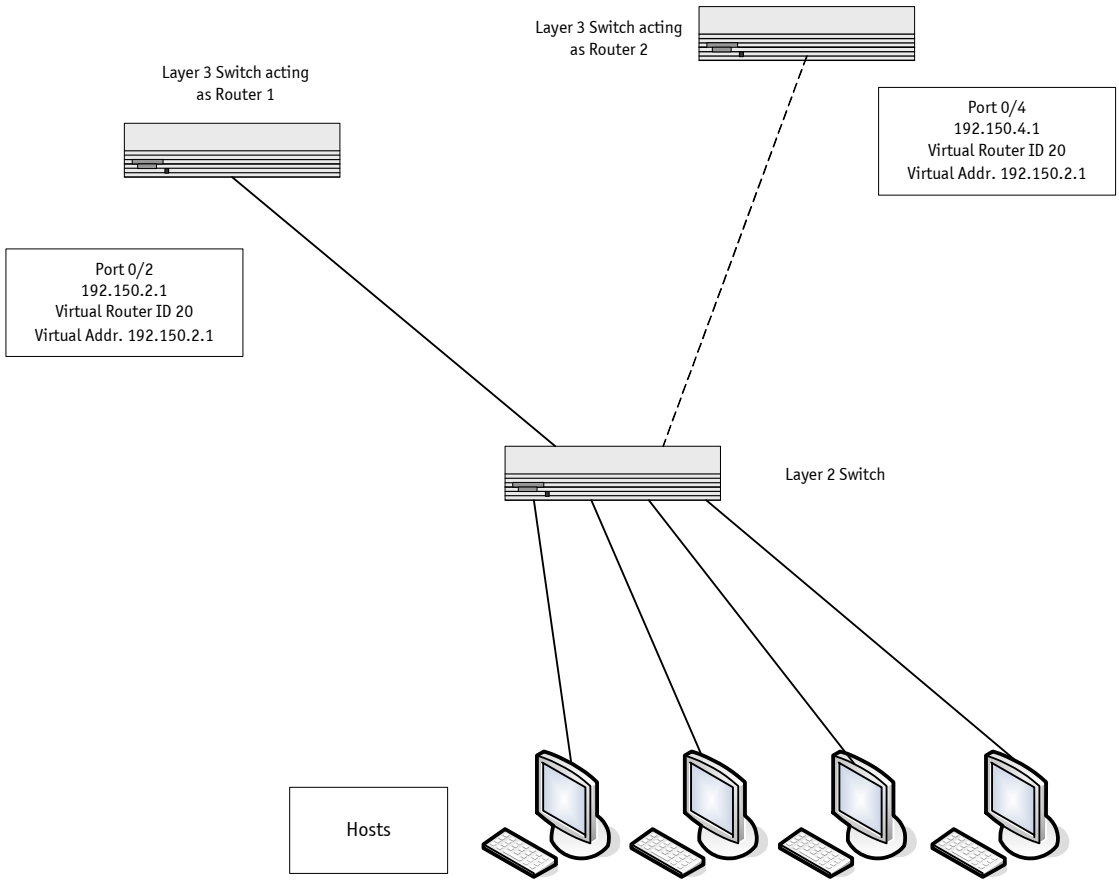
When an end station is statically configured with the address of the router that will handle its routed traffic, a single point of failure is introduced into the network. If the router goes down, the end station is unable to communicate. Since static configuration is a convenient way to assign router addresses, Virtual Router Redundancy Protocol (VRRP) was developed to provide a backup mechanism.

VRRP eliminates the single point of failure associated with static default routes by enabling a backup router to take over from a “master” router without affecting the end stations using the route. The end stations will use a “virtual” IP address that will be recognized by the backup router if the master router fails. Participating routers use an election protocol to determine which router is the master router at any given time. A given port may appear as more than one virtual router to the network, also, more than one port on a FASTPATH software may be configured as a virtual router. Either a physical port or a routed VLAN may participate.

3.3.1 CLI Examples

This example shows how to configure the FASTPATH software to support VRRP. Router 1 will be the default master router for the virtual route, and Router 2 will be the backup router.

Figure 3-4: VRRP Example Network Configuration



3.3.1.1 Example 1: Configuring VRRP on FASTPATH as a Master Router

Enable routing for the switch. IP forwarding is then enabled by default.

```
(Ethernet Fabric) #configure
(Ethernet Fabric) (Config) #ip routing
(Ethernet Fabric) (Config) #exit
```

Configure the IP addresses and subnet masks for the port that will participate in the protocol:

```
(Ethernet Fabric) #configure
(Ethernet Fabric) (Config) #interface 0/2
(Ethernet Fabric) (Config) (interface 0/2) #routing
(Ethernet Fabric) (Config) (interface 0/2) #ip address 192.150.2.1 255.255.255.0
(Ethernet Fabric) (Config) (interface 0/2) #exit
```

Enable VRRP for the switch:

```
(Ethernet Fabric) #configure
(Ethernet Fabric) (Config) #ip vrrp
(Ethernet Fabric) (Config) #exit
```

Assign virtual router IDs to the port that will participate in the protocol:

```
(Ethernet Fabric) #configure
(Ethernet Fabric) (Config) #interface 0/2
(Ethernet Fabric) (Config) (interface 0/2) #ip vrrp 20
```

Specify the IP address that the virtual router function will recognize. Note that the virtual IP address on port 0/2 is the same as the port's actual IP address, therefore this router will always be the VRRP master when it is active. The priority default is 255.

```
(Ethernet Fabric) (Config) (interface 0/2) #ip vrrp 20 ip 192.150.2.1
```

Enable VRRP on the port:

```
(Ethernet Fabric) (Config) (interface 0/2) #ip vrrp 20 mode
(Ethernet Fabric) (Config) (interface 0/2) #exit
```

3.3.1.2 Example 2: Configuring VRRP on FASTPATH as a Backup Router

Enable routing for the switch. IP forwarding is then enabled by default.

```
(Ethernet Fabric) #config
(Ethernet Fabric) (Config) #ip routing
(Ethernet Fabric) (Config) #exit
```

Configure the IP addresses and subnet masks for the port that will participate in the protocol:

```
(Ethernet Fabric) #config
(Ethernet Fabric) (Config) #interface 0/4
(Ethernet Fabric) (Config) (interface 0/4) #routing
(Ethernet Fabric) (Config) (interface 0/4) #ip address 192.150.4.1 255.255.255.0
(Ethernet Fabric) (Config) (interface 0/4) #exit
```

Enable VRRP for the switch.

```
(Ethernet Fabric) #config
(Ethernet Fabric) (Config) #ip vrrp 20
(Ethernet Fabric) (Config) #exit
```

Assign virtual router IDs to the port that will participate in the protocol:

```
(Ethernet Fabric) #config
(Ethernet Fabric) (Config) #interface 0/4
(Ethernet Fabric) (Config) (interface 0/4) #ip vrrp 20
```

Specify the IP address that the virtual router function will recognize. Since the virtual IP address on port 0/4 is the same as Router 1's port 0/2 actual IP address, this router will always be the VRRP backup when Router 1 is active.

```
(Ethernet Fabric) (Config) (interface 0/4) #ip vrrp 20 ip 192.150.2.1
```

Set the priority for the port. The default priority is 100.

```
(Ethernet Fabric) (Config) (interface 0/4) #ip vrrp 20 priority 254
```

Enable VRRP on the port.

```
(Ethernet Fabric) (Config) (interface 0/4) #ip vrrp 20 mode
(Ethernet Fabric) (Config) (interface 0/4) #exit
```

3.4 Proxy Address Resolution Protocol (ARP)

This section describes the Proxy Address Resolution Protocol (ARP) feature.

3.4.1 Overview

- Proxy ARP allows a router to answer ARP requests where the target IP address is not the router itself but a destination that the router can reach.
- If a host does not know the default gateway, proxy ARP can learn the first hop.
- Machines in one physical network appear to be part of another logical network.
- Without proxy ARP, a router responds to an ARP request only if the target IP address is an address configured on the interface where the ARP request arrived.

3.4.2 CLI Examples

The following are examples of the commands used in the proxy ARP feature.

3.4.2.1 Example #1 show ip interface

```
(Ethernet Fabric) #show ip interface ?

<slot/port>              Enter interface in slot/port format.
brief                    Display summary information about IP configuration
                         settings for all ports.

(Ethernet Fabric) #show ip interface 0/24

Routing Mode..................................... Disable
Administrative Mode.............................. Enable
Forward Net Directed Broadcasts.................. Disable
Proxy ARP........................................ Enable
Active State..................................... Inactive
Link Speed Data Rate............................. Inactive
MAC Address...................................... 00:10:18:82:06:5F
Encapsulation Type............................... Ethernet
IP MTU........................................... 1500
```

3.4.2.2 Example #2: Enabling ip proxy-arp

To enable IP Proxy ARP:

```
(Ethernet Fabric) #config
(Ethernet Fabric) [Config) #interface 0/24
(Ethernet Fabric) [Config) (interface 0/24) #ip proxy-arp
(Ethernet Fabric) [Config) (interface 0/24) #exit
```

3.5 OSPF

Larger networks typically use the Open Shortest Path First (OSPF) protocol instead of RIP. To the administrator of a large and/or complex network, OSPF offers several benefits:

- Less network traffic:
    - Routing table updates are sent only when a change has occurred.
    - Only the part of the table that has changed is sent.
    - Updates are sent to a multicast, not a broadcast, address.
- Hierarchical management: allows the network to be subdivided.

FASTPATH supports OSPFv2, which is used on IPv4 networks and OSPFv3, which has enhancements for handling 128-bit IPv6 addresses. The protocols are configured separately within FASTPATH software, but their functionality is largely similar for IPv4 and IPv6 networks. The following description applies to both protocols, except where noted.

3.5.1 OSPF Concepts and Terms

Figure 3-5, Figure 3-63-6, and Figure 3-7 show example OSPF topologies that illustrate the concepts described in this section.

3.5.1.1 Areas and Topology

The top level of the hierarchy of an OSPF network is known as an autonomous system (AS) or routing domain, and is a collection of networks with a common administration and routing strategy. The AS is divided into *areas*. Routers within an area must share detailed information on the topology of their area, but require less detailed information about the topology of other areas. Segregating a network into areas enables limiting the amount of route information communicated throughout the network.

Areas are identified by a numeric ID in IP address format *n.n.n.n* (note, however, that these are not used as actual IP addresses). For simplicity, the area can be configured and referred to in normal integer notation; however, the software converts these to dot notation by using the right-most octet up to 255 and proceeding to the next left octet for higher values (i.e., Area 20 is identified as 0.0.0.20 and Area 256 as 0.0.1.0). The area identified as 0.0.0.0 is referred to as *Area 0* and is considered the *OSPF backbone*. All other OSPF areas in the network must connect to Area 0 directly or through a virtual link. The backbone area is responsible for distributing routing information between non-backbone areas.

A *virtual link* can be used to connect an area to Area 0 when a direct link is not possible. A virtual link traverses an area between the remote area and Area 0 (see Figure 3-7).

A *stub area* is an area that does not receive routes that were learned from a protocol other than OSPF or were statically configured. These routes typically send traffic outside the AS. Therefore, routes from a stub area to locations outside the AS use the default gateway. A virtual link cannot be configured across a stub area. A *Not So Stubby Area* can import limited external routes only from a connected ASBR.

3.5.1.2 OSPF Routers and LSAs

OSPF routers keep track of the state of the various links they send data to. Routers share OSPF *link state advertisements* (LSAs) with other routers. Various LSA types provide detailed information on a link for sharing within an area or summary information for sharing outside an area. External LSAs provide information on static routes or routes learned from other routing protocols.

OSPF defines various router types:

- *Backbone routers* have an interface in Area 0. They condense and summarize information about all the areas in the AS and advertise this information on the backbone.
- *Area border routers (ABRs)* connect areas to the OSPF backbone (in the case of virtual links, the an ABR may connect to another ABR that provides a direct connection to Area 0). An ABR is a member of each area it connects to.
- *Internal routers (IRs)* route traffic within an area. When two routers in an area discover each other through OSPF Hello messages, they are called *OSPF neighbors*. Neighbors share detailed information on the topology of the area using local LSAs.
- *Autonomous system boundary routers (ASBRs)* connect to other ASes. ASBRs use other protocols such as RIP to communicate outside the AS. The ASBR performs *route redistribution*; i.e., when it learns routes from other protocols, it originates external LSAs that advertise those prefixes within the AS.

3.5.1.3 Metrics and Route Selection

You can configure the metric type of external routes originated through route redistribution. The metric type influences the routes computed by other OSPF routers in the domain.

OSPF determines the best route using the assigned cost and the type of the OSPF route. The following order is used for choosing a route if more than one type of route exists:

1. Intra-area (the source and destination address are in the same area)
2. Inter-area (the source and destination are not in the same area, i.e., the route crosses the OSPF backbone)
3. External Type 1
4. External Type 2

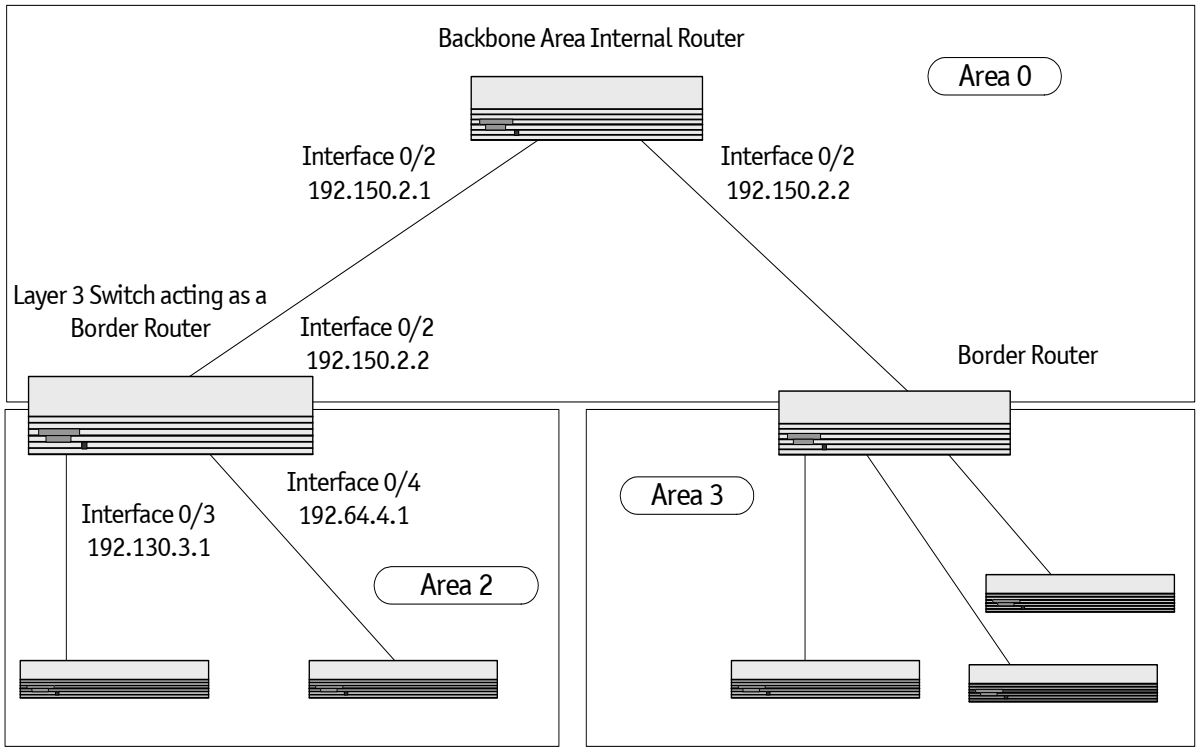
External routes are those imported into OSPF from other routing protocol or processes. OSPF computes the path cost differently for external type 1 and external type 2 routes. The cost of an external type 1 route is the cost advertised in the external LSA plus the path cost from the calculating router to the ASBR. The cost of an external type 2 route is the cost advertised by the ASBR in its external LSA.

3.5.2 CLI Examples

3.5.2.1 Example 1: Configuring an OSPF Border Router and Setting Interface Costs

The following example shows you how to configure an OSPF border router areas and interfaces in FASTPATH software.

Figure 3-5: OSPF Example Network Diagram: Border Router



IPv4 (OSPFv2)	IPv6 (OSPFv3)
• Enable routing for the switch:	
(Ethernet Fabric) #config (Ethernet Fabric) (Config) #ip routing (Ethernet Fabric) (Config) #exit	(Ethernet Fabric) #config (Ethernet Fabric) (Config) #ipv6 unicast-routing (Ethernet Fabric) (Config) #exit
• Enable routing and assign IP for ports 0/2, 0/3, and 0/4.	
(Ethernet Fabric) (Config) #	(Ethernet Fabric) (Config) #
(Ethernet Fabric) (Config) #interface 0/2 routing ip address 192.150.2.2 255.255.255.0 exit	(Ethernet Fabric) (Config) #interface 0/2 routing ipv6 enable exit
(Ethernet Fabric) (Config) #interface 0/3 routing ip address 192.130.3.1 255.255.255.0 exit	(Ethernet Fabric) (Config) #interface 0/3 routing ipv6 address 2002::1/64 exit
(Ethernet Fabric) (Config) #interface 0/4 routing ip address 192.64.4.1 255.255.255.0 exit	(Ethernet Fabric) (Config) #interface 0/4 routing ipv6 address 2003::1/64 exit
(Ethernet Fabric) (Config) #exit	(Ethernet Fabric) (Config) #exit

IPv4 (OSPFv2)	IPv6 (OSPFv3)
• Specify a router ID. Disable 1583 compatibility to prevent a routing loop (IPv4-only).	
(Ethernet Fabric) #config (Ethernet Fabric) (Config) #router ospf (Ethernet Fabric) (Config-router) #router-id 192.150.9.9 (Ethernet Fabric) (Config-router) #no 1583compatibility (Ethernet Fabric) (Config-router) #exit (Ethernet Fabric) (Config) #exit	(Ethernet Fabric) #config (Ethernet Fabric) (Config) #ipv6 router ospf (Ethernet Fabric) (Config-rtr) #router-id 1.1.1.1 (Ethernet Fabric) (Config-rtr) #exit (Ethernet Fabric) (Config) #exit
• OSPF is globally enabled by default. To make it operational on the router, you configure OSPF for particular interfaces and identify which area the interface is associated with. The following commands also sets the priority and cost for the ports:	
(Ethernet Fabric) #config (Ethernet Fabric) (Config) #interface 0/2 ip ospf area 0.0.0.0 ip ospf priority 128 ip ospf cost 32 exit (Ethernet Fabric) (Config) #interface 0/3 ip ospf area 0.0.0.2 ip ospf priority 255 ip ospf cost 64 exit (Ethernet Fabric) (Config) #interface 0/4 ip ospf area 0.0.0.2 ip ospf priority 255 ip ospf cost 64 exit (Ethernet Fabric) (Config) #exit	(Ethernet Fabric) #config (Ethernet Fabric) (Config) #interface 0/2 ipv6 ospf ipv6 ospf areaid 0.0.0.0 ipv6 ospf priority 128 ipv6 ospf cost 32 exit (Ethernet Fabric) (Config) #interface 0/3 ipv6 ospf ipv6 ospf areaid 0.0.0.2 ipv6 ospf priority 255 ipv6 ospf cost 64 exit (Ethernet Fabric) (Config) #interface 0/4 ipv6 ospf ipv6 ospf areaid 0.0.0.2 ipv6 ospf priority 255 ipv6 ospf cost 64 exit (Ethernet Fabric) (Config) #exit

Note...

In OSPFv2, you can also enable OSPF on an interface in global configuration mode by associating a network interface, identified by a network IP address and wildcard mask, with an area. The following example is equivalent to defining interface 0/4 in area 2, as in the previous example:

(Ethernet Fabric) #config

(Ethernet Fabric) (Config) #router ospf

(Ethernet Fabric) (Config) #network 192.164.4.0 0.0.0.255 area 2

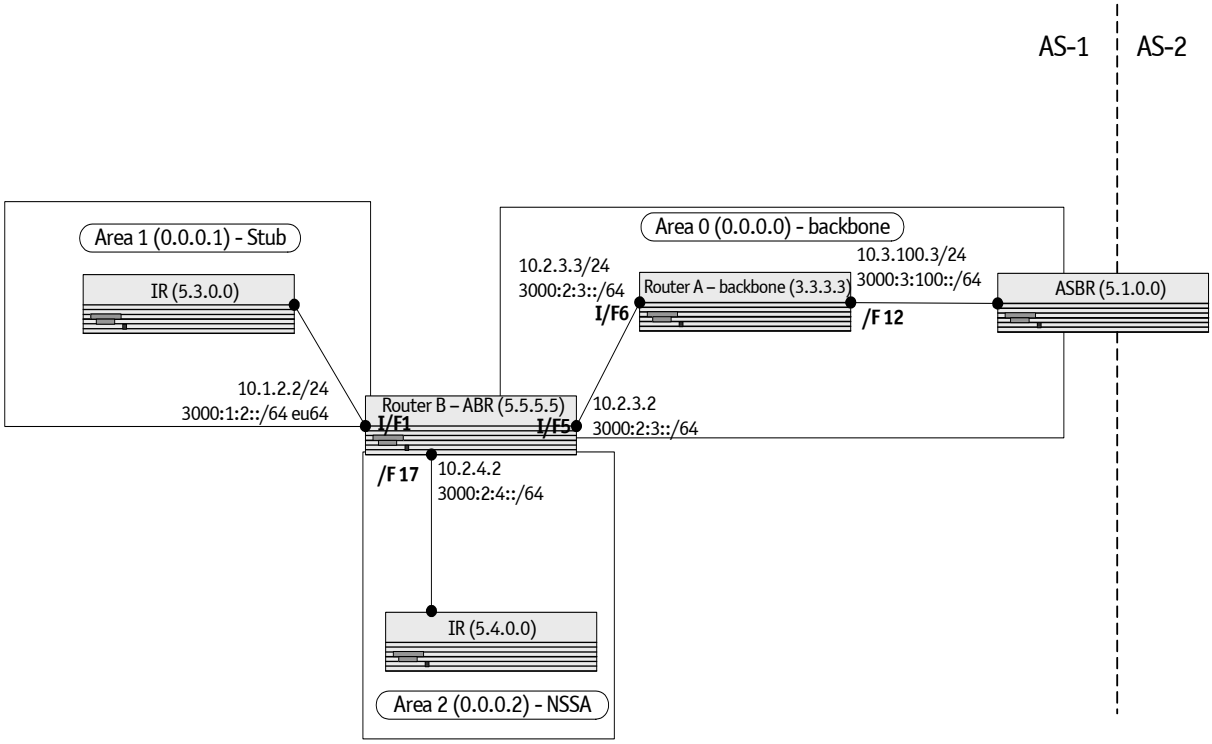
3.5.2.2 Example 2: Configuring Stub and NSSA Areas

In this example, Area 0 connects directly to two other areas: Area 1 is defined as a stub area and Area 2 is defined as an NSSA area.

> **Note...**
>
> OSPFv2 and OSPFv3 can operate concurrently on a network and on the same interfaces (although they do not interact). This example configures both protocols simultaneously.

The following figure illustrates this example OSPF configuration.

Figure 3-6: OSPF Configuration—Stub Area and NSSA Area



Configure Router A: Router A is a backbone router. It links to an ASBR (not defined here) that routes traffic outside the AS.

Globally enable IPv6 and IPv4 routing:

```
(Ethernet Fabric) #configure
(Ethernet Fabric) (Config) #ipv6 unicast-routing
(Ethernet Fabric) (Config) #ip routing
```

Configure IP address and enable OSPF on interfaces 6 and 12 and enable IPv6 OSPF on the interfaces. (OSPF is enabled on the IPv4 interface in the next code group.)

```
(Ethernet Fabric) (Config) #interface 0/6
(Ethernet Fabric) (Config) (interface 0/6) #routing
(Ethernet Fabric) (Config) (interface 0/6) #ip address 10.2.3.3  255.255.255.0
(Ethernet Fabric) (Config) (interface 0/6) #ipv6 address 3000:2:3::/64 eui64
(Ethernet Fabric) (Config) (interface 0/6) #ipv6 ospf
(Ethernet Fabric) (Config) (interface 0/6) #exit
```

```
(Ethernet Fabric) (Config) #interface 0/12
(Ethernet Fabric) (Config) (interface 0/12) #routing
(Ethernet Fabric) (Config) (interface 0/12) #ip address  10.3.100.3  255.255.255.0
(Ethernet Fabric) (Config) (interface 0/12) #ipv6 address 3000:3:100::/64 eui64
(Ethernet Fabric) (Config) (interface 0/12) #ipv6 ospf
(Ethernet Fabric) (Config) (interface 0/12) #exit
```

Define an OSPF router. Enable OSPF for IPv4 on the two interfaces by globally defining the range of IP addresses associated with each interface, and then associating those ranges with Area 0:

```
(Ethernet Fabric) (Config) #ipv6 router ospf
(Ethernet Fabric) (Config-rtr) #router-id 3.3.3.3
(Ethernet Fabric) (Config-rer) #exit
(Ethernet Fabric) (Config) #router ospf
(Ethernet Fabric) (Config-router) #router-id 3.3.3.3
(Ethernet Fabric) (Config-router) #network 10.2.3.0 0.0.0.255 area 0.0.0.0
(Ethernet Fabric) (Config-router) #network 10.3.100.0 0.0.0.255 area 0.0.0.0
(Ethernet Fabric) (Config-router) #exit
(Ethernet Fabric) (Config) #exit
```

Configure Router B: Router B is a ABR that connects Area 0 to Areas 1 and 2.

Configure IPv6 and IPv4 routing. The static routes are included for illustration only: Redistributed static routes, like routes distributed from other protocols, are not injected into stub areas such as Area 1:

```
(Ethernet Fabric) #configure
(Ethernet Fabric) (Config) #ipv6 unicast-routing
(Ethernet Fabric) (Config) #ipv6 route 3000:44:44::/64 3000:2:3::210:18ff:fe82:c14
(Ethernet Fabric) (Config) #ip route 10.23.67.0 255.255.255.0 10.2.3.3
```

On interfaces 1, 5, and 17, configure IPv4 and IPv6 addresses and enable OSPF on the interfaces. For IPv6, associate interface 1 with Area 1 and interface 17 with Area 2. (OSPF is enabled on the IPv4 interface in the next code group.)

```
(Ethernet Fabric) (Config) #interface 0/1
(Ethernet Fabric) (Config) (interface 0/1) #routing
(Ethernet Fabric) (Config) (interface 0/1) #ip address  10.1.2.2  255.255.255.0
(Ethernet Fabric) (Config) (interface 0/1) #ipv6 address 3000:1:2::/64 eui64
(Ethernet Fabric) (Config) (interface 0/1) #ipv6 ospf
(Ethernet Fabric) (Config) (interface 0/1) #ipv6 ospf areaid 1
(Ethernet Fabric) (Config) (interface 0/1) #exit
(Ethernet Fabric) (Config) #interface 0/5
(Ethernet Fabric) (Config) (interface 0/5) #routing
(Ethernet Fabric) (Config) (interface 0/5) #ip address  10.2.3.2  255.255.255.0
(Ethernet Fabric) (Config) (interface 0/5) #ipv6 address 3000:2:3::/64 eui64
(Ethernet Fabric) (Config) (interface 0/5) #ipv6 ospf
(Ethernet Fabric) (Config) (interface 0/5) #exit
(Ethernet Fabric) (Config) #interface 0/17
(Ethernet Fabric) (Config) (interface 0/17) #routing
(Ethernet Fabric) (Config) (interface 0/17) #ip address  10.2.4.2  255.255.255.0
(Ethernet Fabric) (Config) (interface 0/17) #ipv6 address 3000:2:4::/64 eui64
(Ethernet Fabric) (Config) (interface 0/17) #ipv6 ospf
(Ethernet Fabric) (Config) (interface 0/17) #ipv6 ospf areaid 2
(Ethernet Fabric) (Config) (interface 0/17) #exit
```

For IPv4: Define an OSPF router. Define Area 1 as a stub. Enable OSPF for IPv4 on interfaces 1, 5, and 17 by globally defining the range of IP addresses associated with each interface, and then associating those ranges with Areas 1, 0, and 17, respectively. Then, configure a metric cost to associate with static routes when they are redistributed via OSPF:

```
(Ethernet Fabric) (Config) #router ospf
(Ethernet Fabric) (Config-router) #router-id 2.2.2.2
(Ethernet Fabric) (Config-router) #area 0.0.0.1 stub
(Ethernet Fabric) (Config-router) #area 0.0.0.2 nssa
(Ethernet Fabric) (Config-router) #network 10.1.2.0 0.0.0.255 area 0.0.0.1
```

```
(Ethernet Fabric) (Config-router) #network 10.2.3.0 0.0.0.255 area 0.0.0.0
(Ethernet Fabric) (Config-router) #network 10.2.4.0 0.0.0.255 area 0.0.0.2
(Ethernet Fabric) (Config-router) #redistribute static metric 1 subnets
(Ethernet Fabric) (Config-router) #exit
```

For IPv6: Define an OSPF router. Define Area 1 as a stub and area 2 as a Not-So-Stubby-Area (NSSA). Configure a metric cost to associate with static routes when they are redistributed via OSPF:

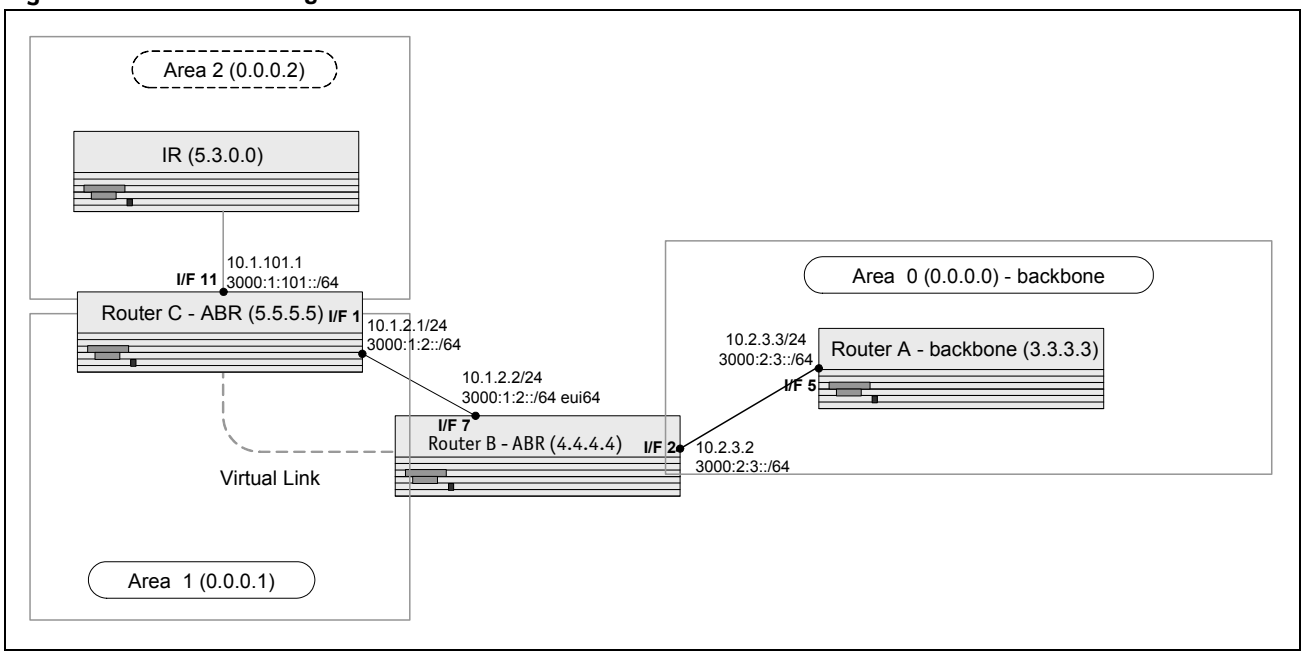
```
(Ethernet Fabric) (Config) #ipv6 router ospf
(Ethernet Fabric) (Config-rtr) #router-id 2.2.2.2
(Ethernet Fabric) (Config-rtr) #area 0.0.0.1 stub
(Ethernet Fabric) (Config-rtr) #area 0.0.0.2 nssa
(Ethernet Fabric) (Config-rtr) #redistribute static metric 105 metric-type 1
(Ethernet Fabric) (Config-rtr) #exit
(Ethernet Fabric) (Config) #exit
```

3.5.2.3 Example 3: Configuring a Virtual Link

In this example, Area 0 connects directly to Area 1. A virtual link is defined that traverses Area 1 and connects to Area 2.

The following figure illustrates this example OSPF configuration.

Figure 3-7: OSPF Configuration—Virtual Link



Configure Router A: Router A is a backbone router. Configuration steps are similar to those for Router A in the previous example.

```
(Ethernet Fabric) #configure
(Ethernet Fabric) (Config) #ipv6 unicast-routing
(Ethernet Fabric) (Config) #ip routing
(Ethernet Fabric) (Config) #exit

(Ethernet Fabric) (Config) #ipv6 router ospf
(Ethernet Fabric) (Config-rtr) #router-id 3.3.3.3
(Ethernet Fabric) (Config-rtr) #exit

(Ethernet Fabric) (Config) #interface 0/5
(Ethernet Fabric) (Config) (interface 0/5) #routing
(Ethernet Fabric) (Config) (interface 0/5) #ip address  10.2.3.3  255.255.255.0
```

```
(Ethernet Fabric) (Config) (interface 0/5) #ipv6 address 3000:2:3::/64 eui64
(Ethernet Fabric) (Config) (interface 0/5) #ipv6 ospf
(Ethernet Fabric) (Config) (interface 0/5) #exit

(Ethernet Fabric) (Config) #router ospf
(Ethernet Fabric) (Config-router) #router-id 3.3.3.3
(Ethernet Fabric) (Config-router) #network 10.2.3.0 0.0.0.255 area 0.0.0.0
(Ethernet Fabric) (Config-router) #exit
(Ethernet Fabric) (Config) #exit
```

Configure Router B: Router B is a ABR that directly connects Area 0 to Area 1. In addition to the configuration steps described in the previous example, we define a virtual link that traverses Area 1 to Router C (5.5.5.5).

```
(Ethernet Fabric) #configure
(Ethernet Fabric) (Config) #ipv6 unicast-routing
(Ethernet Fabric) (Config) ip routing

(Ethernet Fabric) (Config) #interface 0/2
(Ethernet Fabric) (Config) (interface 0/2) #routing
(Ethernet Fabric) (Config) (interface 0/2) #ip address  10.2.3.2  255.255.255.0
(Ethernet Fabric) (Config) (interface 0/2) #ipv6 address 3000:2:3::/64 eui64
(Ethernet Fabric) (Config) (interface 0/2) #ipv6 ospf
(Ethernet Fabric) (Config) (interface 0/2) #exit

(Ethernet Fabric) (Config) #interface 0/7
(Ethernet Fabric) (Config) (interface 0/7) #routing
(Ethernet Fabric) (Config) (interface 0/7) #ip address  10.1.2.2  255.255.255.0
(Ethernet Fabric) (Config) (interface 0/7) #ipv6 address 3000:1:2::211:88FF:FE2A:3CB3/
64 eui64
(Ethernet Fabric) (Config) (interface 0/7) #ipv6 ospf
(Ethernet Fabric) (Config) (interface 0/7) #ipv6 ospf areaid 1
(Ethernet Fabric) (Config) (interface 0/7) #exit

(Ethernet Fabric) (Config) #router ospf
(Ethernet Fabric) (Config-router) #router-id 4.4.4.4
(Ethernet Fabric) (Config-router) #area 0.0.0.1 virtual-link 5.5.5.5
(Ethernet Fabric) (Config-router) #network 10.2.3.0 0.0.0.255 area 0.0.0.0
(Ethernet Fabric) (Config-router) #network 10.1.2.0 0.0.0.255 area 0.0.0.1
(Ethernet Fabric) (Config-router) #exit

(Ethernet Fabric) (Config) #ipv6 router ospf
(Ethernet Fabric) (Config-rtr) #router-id 4.4.4.4
(Ethernet Fabric) (Config-rtr) #area 0.0.0.1 virtual-link 5.5.5.5
(Ethernet Fabric) (Config-rtr) #exit
(Ethernet Fabric) (Config) #exit
```

Configure Router C: Router C is a ABR that enables a virtual link from the remote Area 2 in the AS to Area 0. In addition to the configuration steps described for Router C in the previous example, we define a virtual link that traverses Area 1 to Router B (4.4.4.4).

```
(Ethernet Fabric) #configure
(Ethernet Fabric) (Config) #ipv6 unicast-routing
(Ethernet Fabric) (Config) #ip routing

(Ethernet Fabric) (Config) #interface 0/1
(Ethernet Fabric) (Config) (interface 0/1) #routing
(Ethernet Fabric) (Config) (interface 0/1) #ip address  10.1.2.1  255.255.255.0
(Ethernet Fabric) (Config) (interface 0/1) #ipv6 address 3000:1:2::/64 eui64
(Ethernet Fabric) (Config) (interface 0/1) #ipv6 ospf
(Ethernet Fabric) (Config) (interface 0/1) #ipv6 ospf areaid 1
(Ethernet Fabric) (Config) (interface 0/1) #exit

(Ethernet Fabric) (Config) #interface 0/11
(Ethernet Fabric) (Config) (interface 0/11) #routing
(Ethernet Fabric) (Config) (interface 0/11) #ip address 10.1.101.1 255.255.255.0
(Ethernet Fabric) (Config) (interface 0/11) #ipv6 address 3000:1:101::/64 eui64
```

```
(Ethernet Fabric) (Config) (interface 0/11) #ipv6 ospf
(Ethernet Fabric) (Config) (interface 0/11) #ipv6 ospf areaid 2
(Ethernet Fabric) (Config) (interface 0/11) #exit
(Ethernet Fabric) (Config) #ipv6 router ospf
(Ethernet Fabric) (Config-rtr) #router-id 5.5.5.5
(Ethernet Fabric) (Config-rtr) #area 0.0.0.1 virtual-link 4.4.4.4
(Ethernet Fabric) (Config-rtr) #exit
(Ethernet Fabric) (Config) #router ospf
(Ethernet Fabric) (Config-router) #router-id 5.5.5.5
(Ethernet Fabric) (Config-router) #area 0.0.0.1 virtual-link 4.4.4.4
(Ethernet Fabric) (Config-router) #network 10.1.2.0 0.0.0.255 area 0.0.0.1
(Ethernet Fabric) (Config-router) #network 10.1.101.0 0.0.0.255 area 0.0.0.2
(Ethernet Fabric) (Config-router) #exit
(Ethernet Fabric) (Config) #exit
```

3.6 Routing Information Protocol

Routing Information Protocol (RIP) is one of the protocols which may be used by routers to exchange network topology information. It is characterized as an “interior” gateway protocol, and is typically used in small to medium-sized networks.

3.6.1 RIP Configuration

A router running RIP sends the contents of its routing table to each of its adjacent routers every 30 seconds. When a route is removed from the routing table it is flagged as unusable by the receiving routers after 180 seconds, and removed from their tables after an additional 120 seconds.

There are two versions of RIP:

- RIP-1 defined in RFC 1058
    - Routes are specified by IP destination network and hop count
    - The routing table is broadcast to all stations on the attached network
- RIP-2 defined in RFC 1723
    - Route specification is extended to include subnet mask and gateway
    - The routing table is sent to a multicast address, reducing network traffic
    - An authentication method is used for security

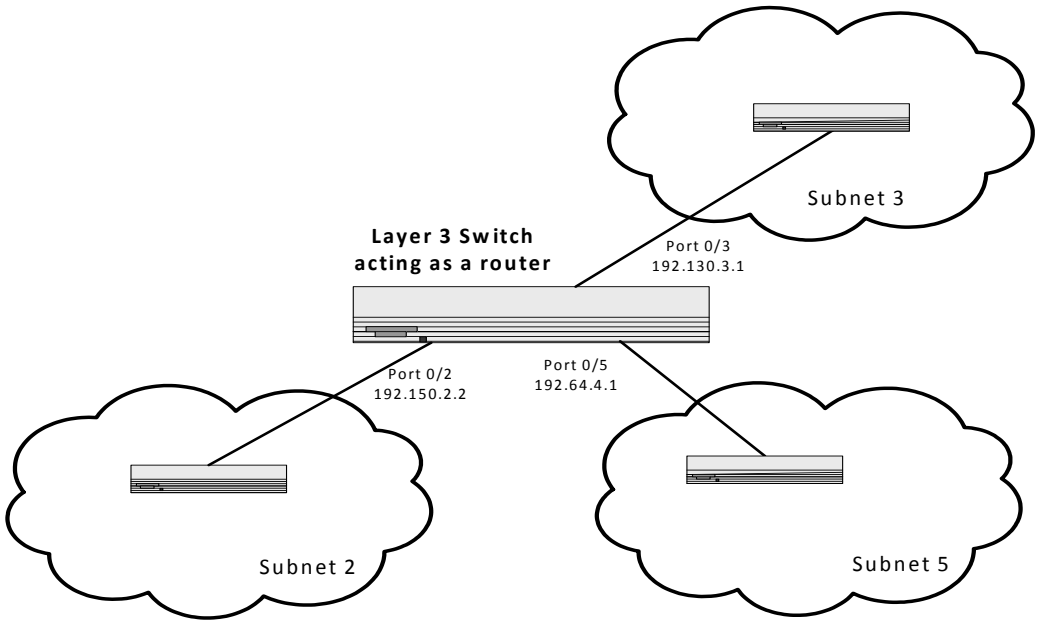
The FASTPATH software supports both versions of RIP. You may configure a given port:

- To receive packets in either or both formats
- To transmit packets formatted for RIP-1 or RIP-2 or to send RIP-2 packets to the RIP-1 broadcast address
- To prevent any RIP packets from being received
- To prevent any RIP packets from being transmitted

3.6.2 CLI Examples

The configuration commands used in the following example enable RIP on ports 0/2 and 0/3 as shown in the network illustrated in Figure 3-8.

Figure 3-8: Port Routing Example Network Diagram



3.6.2.1 Example #1: Enable Routing for the Switch

The following sequence enables routing for the switch:

```
(Ethernet Fabric) #config
(Ethernet Fabric) (Config) #ip routing
(Ethernet Fabric) (Config) #exit
```

3.6.2.2 Example #2: Enable Routing for Ports

The following command sequence enables routing and assigns IP addresses for ports 0/2 and 0/3.

```
(Ethernet Fabric) #config
(Ethernet Fabric) (Config) #interface 0/2
(Ethernet Fabric) (Config) (interface 0/2) #routing
(Ethernet Fabric) (Config) (interface 0/2) #ip address 192.150.2.2 255.255.255.0
(Ethernet Fabric) (Config) (interface 0/2) #exit
(Ethernet Fabric) (Config) #interface 0/3
(Ethernet Fabric) (Config) (interface 0/3) #routing
(Ethernet Fabric) (Config) (interface 0/3) #ip address 192.130.3.1 255.255.255.0
(Ethernet Fabric) (Config) (interface 0/3) #exit
(Ethernet Fabric) (Config) #exit
```

3.6.2.3 Example #3. Enable RIP for the Switch

The next sequence enables RIP for the switch. The route preference defaults to 15.

```
(Ethernet Fabric) #config
(Ethernet Fabric) (Config) #router rip
(Ethernet Fabric) (Config-router) #enable
(Ethernet Fabric) (Config-router) #exit
(Ethernet Fabric) (Config) #exit
```

3.6.2.4 Example #4. Enable RIP for ports 0/2 and 0/3

This command sequence enables RIP for ports 0/2 and 0/3. Authentication defaults to none, and no default route entry is created. The commands specify that both ports receive both RIP-1 and RIP-2 frames, but send only RIP-2 formatted frames.

```
(Ethernet Fabric) #config
(Ethernet Fabric) (Config) #interface 0/2
(Ethernet Fabric) (Config) (interface 0/2) #ip rip
(Ethernet Fabric) (Config) (interface 0/2) #ip rip receive version both
(Ethernet Fabric) (Config) (interface 0/2) #ip rip send version rip2
(Ethernet Fabric) (Config) (interface 0/2) #exit
(Ethernet Fabric) (Config) #interface 0/3
(Ethernet Fabric) (Config) (interface 0/3) #ip rip
(Ethernet Fabric) (Config) (interface 0/3) #ip rip receive version both
(Ethernet Fabric) (Config) (interface 0/3) #ip rip send version rip2
(Ethernet Fabric) (Config) (interface 0/3) #exit
(Ethernet Fabric) (Config) #exit
```

3.7 Route Preferences

You can use route preference assignment to control how the router chooses which routes to use when alternatives exist. This section describes three uses of route preference assignment:

- Assigning Administrative Preferences to Routing Protocols
- Assigning Administrative Preferences to Static Routes
- Using Equal Cost Multipath

3.7.1 Assigning Administrative Preferences to Routing Protocols

The router may learn routes from various sources: static configuration, local route discovery, RIP and OSPF. Most routing protocols use a route metric to determine the shortest path known to the protocol; however, these metrics are independent of one another and not easily comparable. Therefore, when the router learns a route to a particular destination from two different sources, the metrics do not provide a means of choosing the best route for your network.

FASTPATH software enables you to identify the preferred route type by assigning an administrative preference value to each type. The values are arbitrary (1 to 255); however, a route type that has a lower value is preferred over higher-value types.

Local routes are assigned an administrative preference value of 0 and are always preferred over other route types to local hosts. Static routes have a default value of 1; however, this value and all other default preference values are user-configurable.

A protocol can be assigned a preference value of 255 to prevent the router from forwarding packets using that protocol.

3.7.1.1 Example 1

The following commands configure the administrative preference for the RIP:

```
(Ethernet Fabric) #Config
(Ethernet Fabric) (Config) #router rip
(Ethernet Fabric) (Config-router) #distance rip 130
(Ethernet Fabric) (Config-router) #exit
(Ethernet Fabric) #Config
```

For OSPF, an additional parameter identifies the type of OSPF route that the preference value applies to:

```
(Ethernet Fabric) (Config) #router ospf
(Ethernet Fabric) (Config-router) #distance ospf ?

external                Enter preference type external.
inter-area              Enter preference type inter.
intra-area              Enter preference type intra.

(Ethernet Fabric) (Config-router) #distance ospf external 170
(Ethernet Fabric) (Config-router) #exit
```

3.7.1.2 Example 2

By default, static routes are assigned a preference value of 1. The following command changes this default:

```
(Ethernet Fabric) #Config
(Ethernet Fabric) (Config) #ip route distance 20
(Ethernet Fabric) (Config) #exit
```

3.7.2 Assigning Administrative Preferences to Static Routes

When you configure a static route, you can assign a preference value to it. The preference overrides the setting inherited as the default value for static routes.

3.7.2.1 Example 1

In this example, two static routes are defined to the same destination but with different next hops and different preferences (25 and 30). The route with the higher preference will only be used when the preferred route is unavailable:

```
(Ethernet Fabric) #Config
(Ethernet Fabric) (Config) #ip route 10.25.67.0 255.255.255.0 10.25.22.2 25
(Ethernet Fabric) (Config) #ip route 10.25.67.0 255.255.255.0 10.25.21.0 30
(Ethernet Fabric) (Config) #exit
```

3.7.2.2 Example 2

Similarly, you can create two default routes—one preferred and the other used as a backup. In this example, the preference values 1 and 10 are assigned:

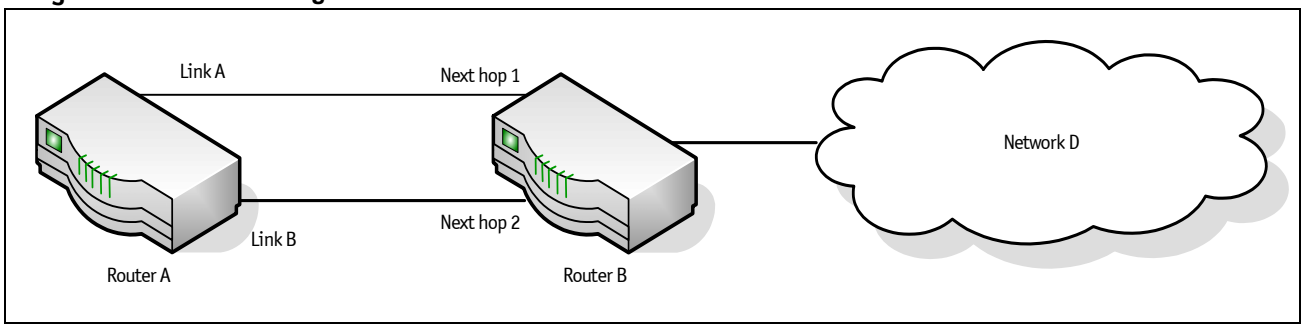
```
(Ethernet Fabric) #Config
(Ethernet Fabric) (Config) #ip route default 10.25.67.2 1
(Ethernet Fabric) (Config) #ip route default 10.25.67.7 10
(Ethernet Fabric) (Config) #exit
```

3.7.3 Using Equal Cost Multipath

The equal cost multipath (ECMP) feature allows a router to use more than one next hop to forward packets to a given destination prefix. It can be used to promote a more optimal use of network resources and bandwidth.

A router that does not use ECMP forwards all packets to a given destination through a single next hop. This next hop may be chosen from among several next hops that provide equally good routes to the destination. For example, in Figure 3-9, Router A sends all traffic to destinations in Network D through next hop NH1, even though the route through NH2 is equally good. Forwarding all traffic via NH1 may cause Link A to be overloaded while Link B is not used at all.

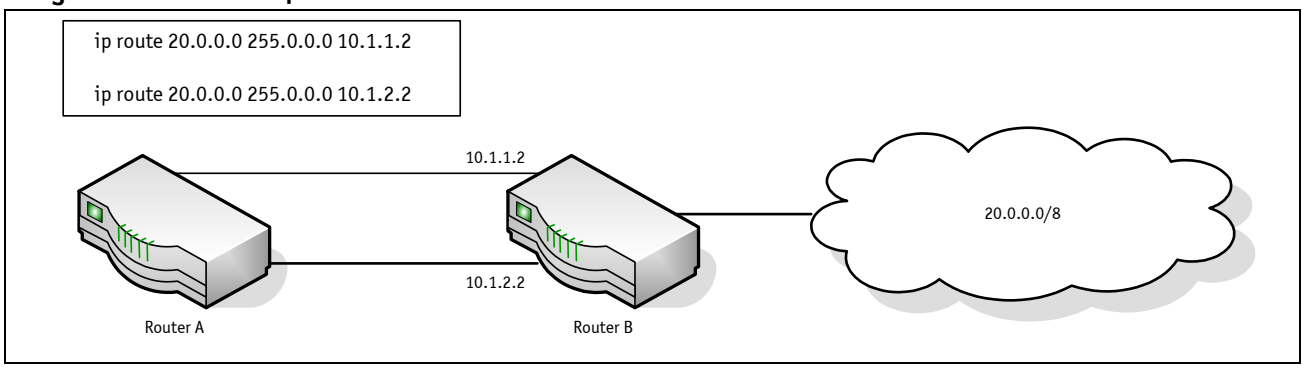
Figure 3-9: Forwarding Without ECMP



With ECMP, Router A can forward traffic to some destinations in Network D via Link A and traffic to other destinations in Network D via Link B, thereby taking advantage of the bandwidth of both links. A hash algorithm is applied to the destination IP addresses to provide a mechanism for selecting among the available ECMP paths.

ECMP routes may be configured statically or learned dynamically. If a user configures multiple static routes to the same destination but with different next hops, then those routes will be treated as a single route with two next hops. For example, given the network in Figure 3-10, if the user configures the following two static routes on Router A, the routing table will contain a single route to 20.0.0.0/8:

Figure 3-10: Next Hop with Two Static Routes



Routing protocols can also be configured to compute ECMP routes. For example, referring to Figure 3-10, if OSPF were configured in on both links connecting Router A and Router B, and if Router B advertised its connection to 20.0.0.0/8, then Router A could compute an OSPF route to 20.0.0.0/8 with next hops of 10.1.1.2 and 10.1.2.2.

Static and dynamic routes are all included in a single combined routing table. This routing table accepts ECMP routes; however, the routing table will not combine routes from different sources to create ECMP routes. Referring to Figure 3-10, assume OSPF is configured on only one of the links between Router A and Router B. Then, on Router A, assume that OSPF reports to the routing table a route to 20.0.0.0/8 with a next hop of 10.1.1.2. If the user also configures a static route to 20.0.0.0/8 with a single next hop of 10.1.2.2, the routing table will **not** combine the OSPF and static routes into a single route to 20.0.0.0/8 with two next hops. All next hops within an ECMP route must be provided by the same source.

An ECMP route contains only next hops whose paths to the destination are of equal cost. Referring to Figure 3-10, if OSPF were configured on all links, but Router A's interface to the 10.1.1.x network had an OSPF link cost of 5 and its interface to the 10.1.2.x network had an OSPF link cost of 10, then OSPF would use only 10.1.1.2 as the next hop to 20.0.0.0/8.

3.7.3.1 Example 1

In the following example, two static routes to the same destination are configured to use different next hops (e.g., for load balancing purposes). Note that the preference metric is not specified, so both routes assume the default static route preference of 1.

```
(Ethernet Fabric) #Config
(Ethernet Fabric) (Config) #ip route 20.0.0.0 255.0.0.0 10.1.1.2
(Ethernet Fabric) (Config) #ip route 20.0.0.0 255.0.0.0 10.1.2.2
(Ethernet Fabric) (Config) #exit
```

The following command adds a third route with a preference value of 5. This route will be used only when the first two are unreachable:

```
(Ethernet Fabric) (Config) #ip route 20.0.0.0 255.0.0.0 10.1.3.2 5
```

3.8 Loopback Interfaces

FASTPATH software provides for the creation, deletion, and management of loopback interfaces.

A loopback interface is a software-only interface that is not associated with a physical location; as such it is not dependent on the physical status of a particular router interface and is always considered “up” as long as the router is running. It enables configuring a stable IP address for remote clients to refer to. The client can communicate with the loopback interface using any available, active router interface.

Note...

In this context, loopback interfaces should not be confused with the loopback IP address, usually 127.0.0.1, assigned to a host for handling self-routed packets.

Loopbacks are typically used for device management purposes. A client can use the loopback interface to communicate with the router through various services such as telnet and SSH. The address on a loopback behaves identically to any of the local addresses of the router in terms of the processing of incoming packets. This interface provides the source address for sent packets and can receive both local and remote packets.

You can create a loopback interface in the Global Config mode by assigning it a unique ID from 0 to 7:

```
(Ethernet Fabric) #Config
(Ethernet Fabric) (Config) #interface loopback 0
```

Next, you assign an IPv4 or IPv6 address to the interface:

```
(Ethernet Fabric) (Config) #interface loopback 0
(Ethernet Fabric) (Config) interface loopback 0 #ip address 192.168.1.2 255.255.255.255
(Ethernet Fabric) (Config) interface loopback 0 #exit
(Ethernet Fabric) (Config) #exit
```

You can view the interface configuration from the Privileged Exec mode:

```
(Ethernet Fabric) #show interface loopback 0

Interface Link Status..................... Up
IP Address................................ 192.168.1.2 255.255.255.255
MTU size.................................. 1500 bytes
```

To delete a loopback interface, enter the following from the Global Config mode:

```
(Ethernet Fabric) #Config
(Ethernet Fabric) (Config) #no interface loopback 0
```

4. Device Security

This chapter describes configuration scenarios for the following features:

- 802.1x Network Access Control
- Access Control Lists (ACLs)
- RADIUS
- TACACS+

4.1 802.1x Network Access Control

Port-based network access control allows the operation of a system's port(s) to be controlled to ensure that access to its services is permitted only by systems that are authorized to do so.

Port Access Control provides a means of preventing unauthorized access by supplicants or users to the services offered by a system. Control over the access to a switch and the LAN to which it is connected can be desirable in order to restrict access to publicly accessible bridge ports or departmental LANs.

FASTPATH achieves access control by enforcing authentication of supplicants that are attached to an authenticator's controlled ports. The result of the authentication process determines whether the supplicant is authorized to access services on that controlled port.

A PAE (Port Access Entity) can adopt one of two roles within an access control interaction:

- Authenticator – Port that enforces authentication before allowing access to services available via that Port.
- Supplicant – Port that attempts to access services offered by the Authenticator.

Additionally, there exists a third role:

- Authentication server – Server that performs the authentication function necessary to check the credentials of the supplicant on behalf of the Authenticator.

Completion of an authentication exchange requires all three roles. FASTPATH supports the authenticator role only, in which the PAE is responsible for communicating with the supplicant. The authenticator PAE is also responsible for submitting information received from the supplicant to the authentication server in order for the credentials to be checked, which determines the authorization state of the port. Depending on the outcome of the authentication process, the authenticator PAE then controls the authorized/unauthorized state of the controlled Port.

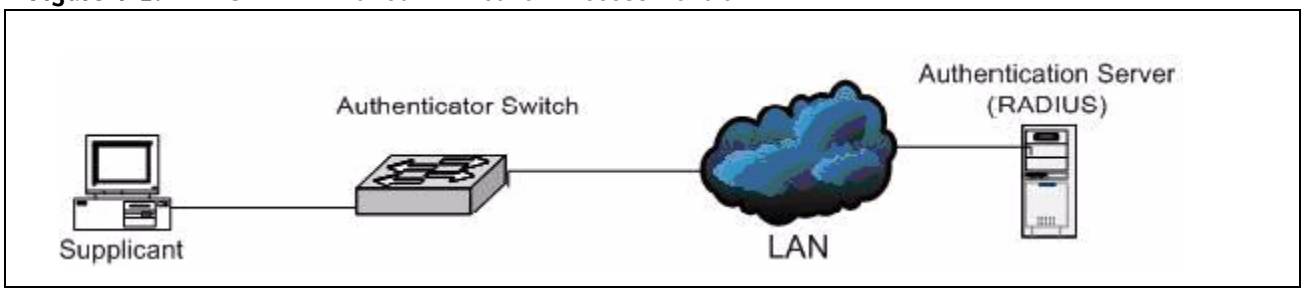
Authentication can be handled locally or via an external authentication server. Two are: Remote Authentication Dial-In User Service (RADIUS) or Terminal Access Controller Access Control System (TACACS+).

RADIUS supports an accounting function to maintain data on service usages. Under RFC 2866, an extension was added to the RADIUS protocol giving the client the ability to deliver accounting information about a user to an accounting server. Exchanges to the accounting server follow similar guidelines as that of an authentication server but the flows are much simpler. At the start of service for a user, the RADIUS client that is configured to use accounting sends an accounting start packet specifying the type of service that it will deliver. Once the server responds with an acknowledgement, the client periodically transmits accounting data. At the end of service delivery, the client sends an accounting stop packet allowing the server to update specified statistics. The server again responds with an acknowledgement.

4.1.1 802.1x Network Access Control Example

This example configures a single RADIUS server used for authentication and accounting at 10.10.10.10. The shared secret is configured to be *secret*. The process creates a new authentication list, called radiusList, which uses RADIUS as the authentication method. This authentication list is associated with the 802.1x default login. 802.1x port based access control is enabled for the system, and interface 0/1 is configured to be in force-authorized mode because this is where the RADIUS server and protected network resources are located.

Figure 4-1: FASTPATH with 802.1x Network Access Control



If a user, or supplicant, attempts to communicate via the switch on any interface except interface 0/1, the system challenges the supplicant for login credentials. The system encrypts the provided information and transmits it to the RADIUS server. If the RADIUS server grants access, the system sets the 802.1x port state of the interface to authorized and the supplicant is able to access network resources.

```
(Ethernet Fabric) #config
(Ethernet Fabric) (Config)#radius server host auth 10.10.10.10
(Ethernet Fabric) (Config)#radius server key auth 10.10.10.10
   Enter secret (16 characters max): secret
   Enter secret (16 characters max): secret
(Ethernet Fabric) (Config)radius server host acct 10.10.10.10
(Ethernet Fabric) (Config)radius server key acct 10.10.10.10
   Enter secret (16 characters max): secret
   Enter secret (16 characters max): secret
(Ethernet Fabric) (Config)radius accounting mode
(Ethernet Fabric) (Config)#authentication login radiusList radius
(Ethernet Fabric) (Config)#dot1x defaultlogin radiusList
(Ethernet Fabric) (Config)#dot1x system-auth-control
(Ethernet Fabric) (Config)#interface 0/1
(Ethernet Fabric) (Config) (interface 0/1)#dot1x port-control force-authorized
(Ethernet Fabric) (Config) (interface 0/1)#exit
(Ethernet Fabric) (Config)#exit
```

4.2 Access Control Lists (ACLs)

This section describes the Access Control Lists (ACLs) feature.

4.2.1 Overview

Access Control Lists (ACLs) are a collection of permit and deny conditions, called rules, that provide security by blocking unauthorized users and allowing authorized users to access specific resources.

ACLs can also provide traffic flow control, restrict contents of routing updates, and decide which types of traffic are forwarded or blocked. Normally ACLs reside in a firewall router or in a router connecting two internal networks.

ACL support features include Flow-based Mirroring and ACL Logging.

- Flow-based mirroring is the ability to mirror traffic that matches a permit rule to a specific physical port or LAG. Flow-based mirroring is similar to the redirect function, except that in flow-based mirroring a copy of the permitted traffic is delivered to the mirror interface while the packet itself is forwarded normally through the device. You cannot configure a given ACL rule with mirror and redirect attributes.
- ACL Logging provides a means for counting the number of “hits” against an ACL rule. When you configure ACL Logging, you augment the ACL deny rule specification with a ‘log’ parameter that enables hardware hit count collection and reporting. FASTPATH uses a fixed five minute logging interval, at which time trap log entries are written for each ACL logging rule that accumulated a non-zero hit count during that interval. You cannot configure the logging interval.

Using ACLs to mirror traffic is called flow-based mirroring since the traffic flow is defined by the ACL classification rules. This is in contrast to port mirroring, where all traffic encountered on a specific interface is replicated on another interface.

You can set up ACLs to control traffic at Layer 2, Layer 3, or Layer 4. MAC ACLs operate on Layer 2. IP ACLs operate on Layers 3 and 4.

4.2.1.1 Limitations

There are limitations with respect to ACLs. These limitations are platform dependent.

- Maximum number of ACLs.
- Maximum rules per ACL.
- The system supports ACLs set up for inbound traffic only.
- You can configure mirror or redirect attributes for a given ACL rule, but not both.
- Some hardware platforms do not support MAC ACLs and IP ACLs on the same interface.
- A hardware platform may support a limited number of counter resources, so it may not be possible to log every ACL rule. You can define an ACL with any number of logging rules, but the number of rules that are actually logged cannot be determined until the ACL is applied to an interface. Furthermore, hardware counters that become available after an ACL is applied are not retroactively assigned to rules that were unable to be logged (the ACL must be un-applied then re-applied). Rules that are unable to be logged are still active in the ACL for purposes of permitting or denying a matching packet.
- The order of the rules is important: when a packet matches multiple rules, the first rule takes precedence. Also, once you define an ACL for a given port, all traffic not specifically permitted by the ACL is denied access.

4.2.2 MAC ACLs

MAC ACLs are Layer 2 ACLs. You can configure the rules to inspect the following fields of a packet (limited by platform):

- Source MAC address
- Source MAC mask
- Destination MAC address
- Destination MAC mask
- VLAN ID
- Class of Service (CoS) (802.1p)
- Ethertype

L2 ACLs can apply to one or more interfaces.

Multiple access lists can be applied to a single interface; sequence number determines the order of execution.

You can assign packets to queues using the assign queue option.

4.2.3 IP ACLs

IP ACLs classify for Layers 3 and 4.

Each ACL is a set of up to ten rules applied to inbound traffic. Each rule specifies whether the contents of a given field should be used to permit or deny access to the network, and may apply to one or more of the following fields within a packet:

- Destination IP with wildcard mask
- Destination L4 Port
- Every Packet
- IP DSCP
- IP Precedence
- IP TOS
- Protocol
- Source IP with wildcard mask
- Source L4 port
- Destination Layer 4 port

4.2.4 ACL Configuration Process

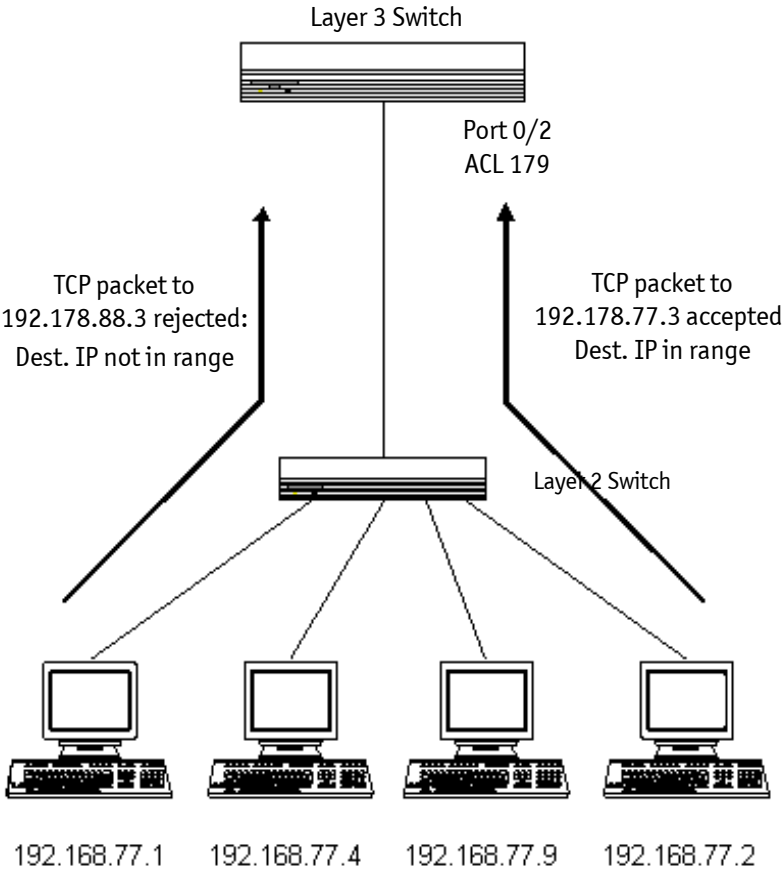
To configure ACLs, follow these steps:

1. Create a MAC ACL by specifying a name.
2. Create an IP ACL by specifying a number.
3. Add new rules to the ACL.
4. Configure the match criteria for the rules.
5. Apply the ACL to one or more interfaces.

4.2.5 IP ACL CLI Examples

The script in this section shows you how to set up an IP ACL with two rules, one applicable to TCP traffic and one to UDP traffic. The content of the two rules is the same. TCP and UDP packets will only be accepted by the FASTPATH switch if the source and destination stations have IP addresses that fall within the defined sets.

Figure 4-2: IP ACL Example Network Diagram



4.2.5.1 Example #1: Create ACL 179 and Define an ACL Rule

After the mask has been applied, it permits packets carrying TCP traffic that matches the specified Source IP address, and sends these packets to the specified Destination IP address.

```
(Ethernet Fabric) #config
(Ethernet Fabric) (Config)#access-list 179 permit tcp 192.168.77.0 0.0.0.255
192.168.77.3 0.0.0.0
```

4.2.5.2 Example #2: Define the Second Rule for ACL 179

Define the rule to set similar conditions for UDP traffic as for TCP traffic.

```
(Ethernet Fabric) (Config)#access-list 179 permit udp 192.168.77.0 0.0.0.255
192.168.77.3 0.0.0.255
Ethernet Fabric) (Config)#exit
```

4.2.5.3 Example #3: Apply the rule to Inbound Traffic on Port 0/2

Only traffic matching the criteria will be accepted.

```
Ethernet Fabric) (Config)#interface 0/2
Ethernet Fabric) (Config)(interface 0/2)#ip access-group 179 in
Ethernet Fabric) (Config)(interface 0/2)#exit
```

4.2.6 MAC ACL CLI Examples

The following are examples of the commands used for the MAC ACLs feature.

4.2.6.1 Example #4: Set up a MAC Access List

```
(Ethernet Fabric) #Config
(Ethernet Fabric) (Config)#mac access-list ?

extended                    Configure extended MAC Access List parameters.

(Ethernet Fabric) (Config)#mac access-list extended ?

<name>                      Enter access-list name up to 31 characters in length.
rename                      Rename MAC Access Control List.

(Ethernet Fabric) (Config)#mac access-list extended mac1 ?

<cr>                        Press Enter to execute the command.

(Ethernet Fabric) (Config)#mac access-list extended mac1
(Ethernet Fabric) (Config)#exit
```

4.2.6.2 Example #5: Specify MAC ACL Attributes

```
(Ethernet Fabric) (Config)#mac access-list extended mac1
(Ethernet Fabric) (Config-mac-access-list)# deny ?

<srcmac>                  Enter a MAC Address.
any                       Configure a match condition for all the source MAC
                          addresses in the Source MAC Address field.
```

```
(Ethernet Fabric) (Config-mac-access-list)#deny any ?

<dstmac>                  Enter a MAC Address.
any                       Configure a match condition for all the destination
                          MAC addresses in the Destination MAC Address field.
bpdu                      Match on any BPDU destination MAC Address.

(Ethernet Fabric) (Config-mac-access-list)#deny any 00:11:22:33:44:55 ?

<dstmacmask>              Enter a MAC Address bit mask.

(Ethernet Fabric) (Config-mac-access-list)#deny any 00:11:22:33:44:55 00:00:00:00:FF:FF
?

<ethertypekey>            Enter one of the following keywords to specify an
                          Ethertype (appletalk, arp, ibmsna, ipv4, ipv6, ipx,
                          mplsmcast, mplsucast, netbios, novell, pppoe, rarp).
<0x0600-0xffff>           Enter a four-digit hexadecimal number in the range of
                          0x0600 to 0xffff to specify a custom Ethertype value.
vlan                      Configure a match condition based on a VLAN ID.
cos                       Configure a match condition based on a COS value.
log                       Configure logging for this access list rule.
assign-queue              Configure the Queue Id assignment attribute.
<cr>                      Press Enter to execute the command.

(Ethernet Fabric) (Config-mac-access-list)#deny any 00:11:22:33:44:55 00:00:00:00:FF:FF
log ?

assign-queue              Configure the Queue Id assignment attribute.
<cr>                      Press Enter to execute the command.

(Ethernet Fabric) (Config-mac-access-list)#deny any 00:11:22:33:44:55 00:00:00:00:FF:FF
log
(Ethernet Fabric) (Config-mac-access-list)#exit
(Ethernet Fabric) (Config)#exit
```

4.2.6.3 Example #6 Configure MAC Access Group

```
(Ethernet Fabric) # Config
(Ethernet Fabric) (Config)# interface 0/5
(Ethernet Fabric) (Config) (interface 0/5)# mac ?

access-group              Attach MAC Access List to Interface.

(Ethernet Fabric) (Config) (interface 0/5)# mac access-group ?

<name>                    Enter name of MAC Access Control List.

(Ethernet Fabric) (Config) (interface 0/5)# mac access-group mac1 ?

in                        Enter the direction <in>.

(Ethernet Fabric) (Config) (interface 0/5)# mac access-group mac1 in ?

<cr>                      Press Enter to execute the command.
<1-4294967295>            Enter the sequence number (greater than 0) to
                          rank direction. A lower sequence number
                          has higher precedence.
```

```
(Ethernet Fabric) (Config) (interface 0/5)# mac access-group mac1 in 6 ?

<cr>                          Press Enter to execute the command.

(Ethernet Fabric) (Config) (interface 0/5)# mac access-group mac1 in 6
(Ethernet Fabric) (Config) (interface 0/5)# exit
(Ethernet Fabric) (Config)# exit
```

4.2.6.4 Example #7: Set up an ACL with Permit Action

```
(Ethernet Fabric) # Config
(Ethernet Fabric) (Config)# mac access-list extended mac2
(Ethernet Fabric) (Config)# permit ?

<srcmac>                  Enter a MAC Address.

any                       Configure a match condition for all the source MAC
                          addresses in the Source MAC Address field.
(Ethernet Fabric) (Config)# permit any ?

<dstmac>                  Enter a MAC Address.

any                       Configure a match condition for all the destination
                          MAC addresses in the Destination MAC Address field.

bpdu                      Match on any BPDU destination MAC Address.

(Ethernet Fabric) (Config)# permit any any?

<ethertypekey>            Enter one of the following keywords to specify an
                          Ethertype (appletalk, arp, ibmsna, ipv4, ipv6, ipx,
                          mplsmcast, mplsucast, netbios, novell, pppoe, rarp).

<0x0600-0xffff>           Enter a four-digit hexadecimal number in the range of
                          0x0600 to 0xffff to specify a custom Ethertype value.

vlan                      Configure a match condition based on a VLAN ID.

cos                       Configure a match condition based on a COS value.

log                       Configure logging for this access list rule.

assign-queue              Configure the Queue Id assignment attribute.

<cr>                      Press Enter to execute the command.

(Ethernet Fabric) (Config)# permit any any
```

4.2.6.5 Example #8: Show MAC Access Lists

```
(Ethernet Fabric) #show mac access-lists
Current number of all ACLs: 2Maximum number of all ACLs: 100

MAC ACL Name Rules Direction Interface(s)
------------ ----- --------- ------------
mac1         1     inbound   0/5
mac2         1

(Ethernet Fabric) #show mac access-lists mac1

MAC ACL Name: mac1
```

```
Rule Number: 1
Action..................................... deny
Destination MAC Address.................... 00:11:22:33:44:55
Destination MAC Mask....................... 00:00:00:00:FF:FF
Log........................................ TRUE
```

4.3 RADIUS

Making use of a single database of accessible information—as in an Authentication Server—can greatly simplify the authentication and management of users in a large network. One such type of Authentication Server supports the Remote Authentication Dial In User Service (RADIUS) protocol as defined by RFC 2865.

For authenticating users prior to access, the RADIUS standard has become the protocol of choice by administrators of large accessible networks. To accomplish the authentication in a secure manner, the RADIUS client and RADIUS server must both be configured with the same shared password or “secret”. This “secret” is used to generate one-way encrypted authenticators that are present in all RADIUS packets. The “secret” is never transmitted over the network.

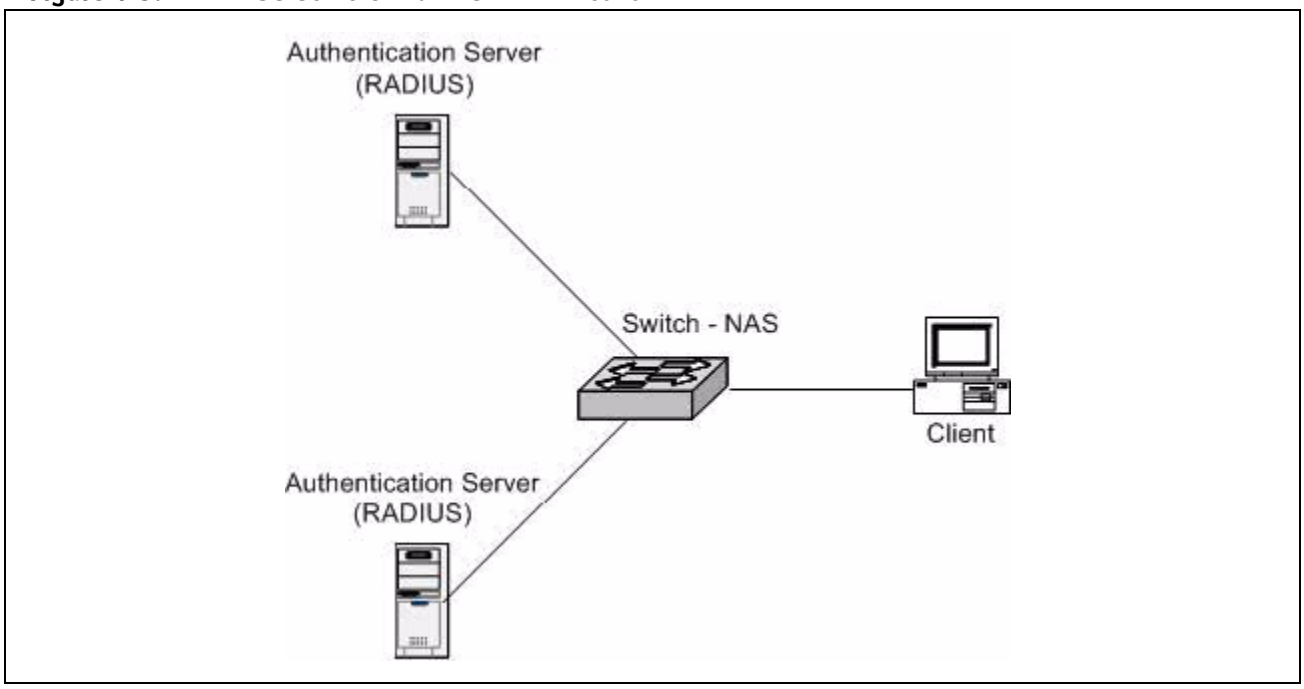
RADIUS conforms to a secure communications client/server model using UDP as a transport protocol. It is extremely flexible, supporting a variety of methods to authenticate and statistically track users. RADIUS is also extensible, allowing for new methods of authentication to be added without disrupting existing functionality.

As a user attempts to connect to a functioning RADIUS supported network, a device referred to as the Network Access Server (NAS) or switch/router first detects the contact. The NAS or user-login interface then prompts the user for a name and password. The NAS encrypts the supplied information and a RADIUS client transports the request to a pre-configured RADIUS server. The server can authenticate the user itself, or make use of a back-end device to ascertain authenticity. In either case a response may or may not be forthcoming to the client. If the server accepts the user, it returns a positive result with attributes containing configuration information. If the server rejects the user, it returns a negative result. If the server rejects the client or the shared “secrets” differ, the server returns no result. If the server requires additional verification from the user, it returns a challenge, and the request process begins again.

4.3.1 RADIUS Configuration Example

This example configures two RADIUS servers at 10.10.10.10 and 11.11.11.11. Each server has a unique shared secret key. The shared secrets are configured to be *secret1* and *secret2* respectively. The server at 10.10.10.10 is configured as the primary server. The process creates a new authentication list, called radiusList, which uses RADIUS as the primary authentication method, and local authentication as a backup method in the event that the RADIUS server cannot be contacted. This authentication list is then associated with the default login.

Figure 4-3: RADIUS Servers in a FASTPATH Network



When a user attempts to log in, the switch prompts for a username and password. The switch then attempts to communicate with the primary RADIUS server at 10.10.10.10. Upon successful connection with the server, the login credentials are exchanged over an encrypted channel. The server grants or denies access, which the switch honors, and either allows or does not allow the user to access the switch. If neither of the two servers can be contacted, the switch searches its local user database for the user.

```
(Ethernet Fabric) # config
(Ethernet Fabric) (Config)#radius server host auth 10.10.10.10
(Ethernet Fabric) (Config)#radius server key auth 10.10.10.10
   Enter secret (16 characters max): secret1
   Enter secret (16 characters max): secret1
(Ethernet Fabric) (Config)#radius server host auth 11.11.11.11
(Ethernet Fabric) (Config)#radius server key auth 11.11.11.11
   Enter secret (16 characters max): secret2
   Enter secret (16 characters max): secret2
(Ethernet Fabric) (Config)#radius server primary 10.10.10.10
(Ethernet Fabric) (Config)#authentication login radiusList radius local
(Ethernet Fabric) (Config)#users defaultlogin radiusList
(Ethernet Fabric) (Config)#exit
```

4.4 TACACS+

TACACS+ (Terminal Access Controller Access Control System) provides access control for networked devices via one or more centralized servers. Similar to RADIUS, this protocol simplifies authentication by making use of a single database that can be shared by many clients on a large network. TACACS+ is based on the TACACS protocol described in RFC1492. TACACS+ uses TCP to ensure reliable delivery and a shared key configured on the client and daemon server to encrypt all messages.

After you configure TACACS+ as the authentication method for user login, the NAS (Network Access Server) prompts for the user login credentials and requests services from the FASTPATH TACACS+ client. The client then uses the configured list of servers for authentication, and provides results back to the NAS. You can configure the TACACS+ server list with one or more hosts defined via their network IP address. You can also assign each a priority to determine the order in which the TACACS+ client will contact them. TACACS+ contacts the server when a connection attempt fails or times out for a higher priority server.

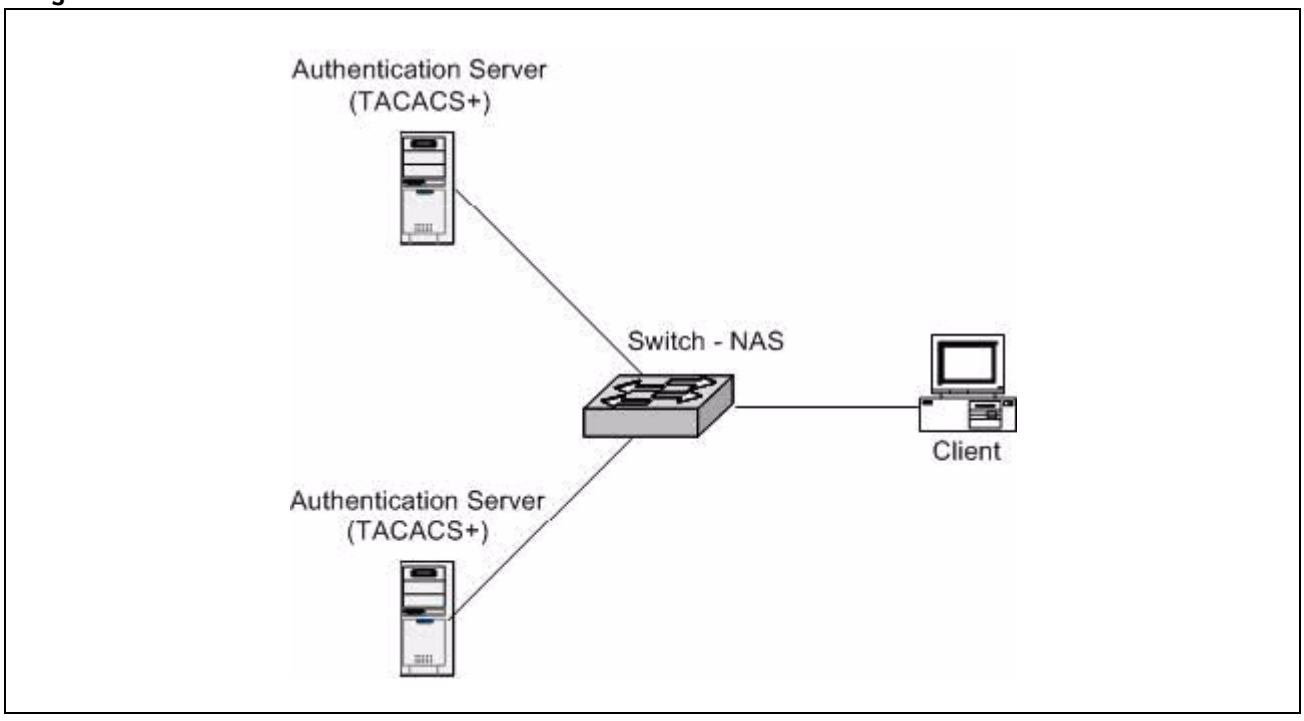
You can configure each server host with a specific connection type, port, timeout, and shared key, or you can use global configuration for the key and timeout.

Like RADIUS, the TACACS+ server can do the authentication itself, or redirect the request to another back-end device. All sensitive information is encrypted and the shared secret is never passed over the network; it is used only to encrypt the data.

4.4.1 TACACS+ Configuration Example

This example configures two TACACS+ servers at 10.10.10.10 and 11.11.11.11. Each server has a unique shared secret key. The server at 10.10.10.10 has a default priority of 0, the highest priority, while the other server has a priority of 2. The process creates a new authentication list, called tacacsList, which uses TACACS+ to authenticate, and uses local authentication as a backup method. This authentication list is then associated with the defaultlogin.

Figure 4-4: FASTPATH with TACACS+



When a user attempts to log into the switch, the NAS or switch prompts for a username and password. The switch attempts to communicate with the highest priority configured TACACS+ server at 10.10.10.10. Upon successful connection with the server, the switch and server exchange the login credentials over an encrypted channel. The server then grants or denies access, which the switch honors, and either allows or does not allow the user to gain access to the switch. If neither of the two servers can be contacted, the switch searches its local user database for the user.

```
(Ethernet Fabric) # config
(Ethernet Fabric) (Config)#tacacs-server host 10.10.10.10
(Ethernet Fabric) (Tacacs)#key tacacs1
(Ethernet Fabric) (Tacacs)#exit
(Ethernet Fabric) (Config)#tacacs-server host 11.11.11.11
(Ethernet Fabric) (Tacacs)#key tacacs2
(Ethernet Fabric) (Tacacs)#priority 2
(Ethernet Fabric) (Tacacs)#exit
(Ethernet Fabric) (Config)#authentication login tacacsList tacacs local
(Ethernet Fabric) (Config)#users defaultlogin tacacsList
(Ethernet Fabric) (Config)#exit
```

5. IPv6

This chapter includes the following sections:

- Overview
- Interface Configuration
- DHCPv6

5.1 Overview

There are many conceptual similarities between IPv4 and IPv6 network operation. Addresses still have a network prefix portion (subnet) and a device interface specific portion (host). While the length of the network portion is still variable, most users have standardized on using a network prefix length of 64 bits. This leaves 64 bits for the interface specific portion, called an Interface ID in IPv6. Depending upon the underlying link addressing, the Interface ID can be automatically computed from the link (e.g., MAC address). Such an automatically computed Interface ID is called an EUI64 identifier.

IPv6 packets on the network are of an entirely different format than traditional IPv4 packets and are also encapsulated in a different EtherType (contained within the L2 header to indicate which L3 protocol is used). In order to route these packets across L3 requires an infrastructure equivalent to and parallel to that provided for IPv4.

Note...

FASTPATH also implements OSPFv3 for use with IPv6 networks. These configuration scenarios are included with the OSPFv2 scenarios in chapter 3, section „OSPF“.

5.2 Interface Configuration

In FASTPATH software, IPv6 coexists with IPv4. As with IPv4, IPv6 routing can be enabled on physical and VLAN interfaces. Each L3 routing interface can be used for IPv4, IPv6, or both.

Neighbor discovery is the IPv6 replacement for Address Resolution Protocol (ARP). Router advertisement is part of the neighbor discovery process and is required for IPv6. As part of router advertisement, FASTPATH software supports stateless auto configuration of end nodes. FASTPATH software supports both EUI-64 interface identifiers and manually configured interface IDs.

While optional in IPv4, router advertisement is mandatory in IPv6. Router advertisements specify the network prefix(es) on a link which can be used by receiving hosts, in conjunction with an EUI64 identifier, to auto configure a host's address. Routers have their network prefixes configured and may use EUI64 or manually configured interface IDs. In addition to one or more global addresses, each IPv6 interface also has an auto-configured link-local address which is:

- Allocated from part of the IPv6 unicast address space
- Not visible off the local link
- Not globally unique

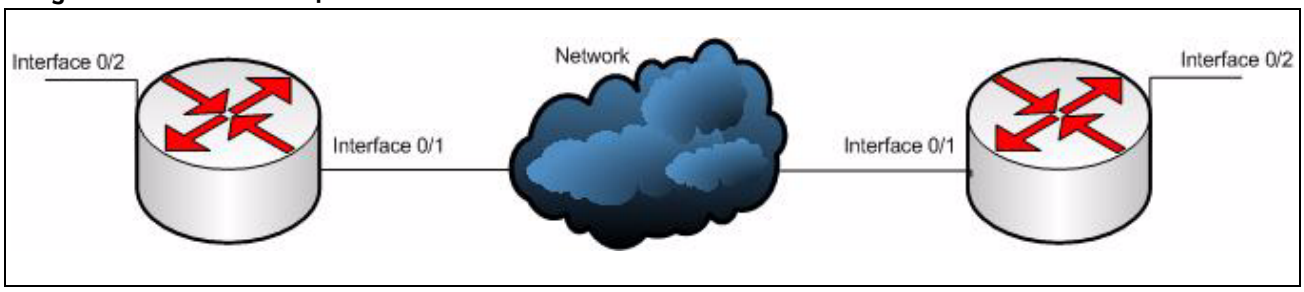
Next hop addresses computed by routing protocols are usually link-local.

During a transition period, a global IPv6 Internet backbone may not be available. The solution of this is to tunnel IPv6 packets inside IPv4 to reach remote IPv6 islands. When a packet is sent over such a link, it is encapsulated in IPv4 in order to traverse an IPv4 network and has the IPv4 headers removed at the other end of the tunnel.

5.2.1 CLI Example

In Figure 5-1, two devices are connected as shown in the diagram. Interface 0/1 on both devices connects to an IPv4 backbone network where OSPF is used as the dynamic routing protocol to exchange IPv4 routes. OSPF allows device 1 and device 2 to learn routes to each other (from the 20.20.20.x network to the 10.10.10.x network and vice versa). Interface 0/2 on both devices connects to the local IPv6 network. OSPFv3 is used to exchange IPv6 routes between the two devices. The tunnel interface allows data to be transported between the two remote IPv6 networks over the IPv4 network.

Figure 5-1: IPv6 Example



Device 1

```
(Ethernet Fabric) # config
(Ethernet Fabric) (Config) #ip routing
(Ethernet Fabric) (Config) #ipv6 unicast-routing
(Ethernet Fabric) (Config) #router ospf
(Ethernet Fabric) (Config-router) #router-id 1.1.1.1
(Ethernet Fabric) (Config-router) #exit

(Ethernet Fabric) (Config) #ipv6 router ospf
(Ethernet Fabric) (Config-rtr) #router-id 1.1.1.1
(Ethernet Fabric) (Config-rtr) #exit

(Ethernet Fabric) (Config) #interface 0/1
(Ethernet Fabric) (Config) (interface 0/1) #routing
(Ethernet Fabric) (Config) (interface 0/1) #ip address 20.20.20.1 255.255.255.0
(Ethernet Fabric) (Config) (interface 0/1) #ip ospf area 0.0.0.0
(Ethernet Fabric) (Config) (interface 0/1) #exit
```

```
(Ethernet Fabric) (Config) #interface 0/2
(Ethernet Fabric) (Config) (interface 0/2) #routing
(Ethernet Fabric) (Config) (interface 0/2) #ipv6 enable
(Ethernet Fabric) (Config) (interface 0/2) #ipv6 address 2020:1::1/64
(Ethernet Fabric) (Config) (interface 0/2) #ipv6 ospf
(Ethernet Fabric) (Config) (interface 0/2) #ipv6 ospf network point-to-point
(Ethernet Fabric) (Config) (interface 0/2) #exit

(Ethernet Fabric) (Config) #interface tunnel 0
(Ethernet Fabric) (Config) (interface tunnel 0) #ipv6 address 2001::1/64
(Ethernet Fabric) (Config) (interface tunnel 0) #tunnel mode ipv6ip
(Ethernet Fabric) (Config) (interface tunnel 0) #tunnel source 20.20.20.1
(Ethernet Fabric) (Config) (interface tunnel 0) #tunnel destination 10.10.10.1
(Ethernet Fabric) (Config) (interface tunnel 0) #ipv6 ospf
(Ethernet Fabric) (Config) (interface tunnel 0) #ipv6 ospf network point-to-point
(Ethernet Fabric) (Config) (interface tunnel 0) #exit

(Ethernet Fabric) (Config) #interface loopback 0
(Ethernet Fabric) (Config) (interface loopback 0) #ip address 1.1.1.1 255.255.255.0
(Ethernet Fabric) (Config) (interface loopback 0) #exit
(Ethernet Fabric) (Config) #exit
```

Device 2

```
(Ethernet Fabric) # config
(Ethernet Fabric) (Config) #ip routing
(Ethernet Fabric) (Config) #ipv6 unicast-routing
(Ethernet Fabric) (Config) #router ospf
(Ethernet Fabric) (Config-router) #router-id 2.2.2.2
(Ethernet Fabric) (Config-router) #exit

(Ethernet Fabric) (Config) #ipv6 router ospf
(Ethernet Fabric) (Config-rtr) #router-id 2.2.2.2
(Ethernet Fabric) (Config-rtr) #exit

(Ethernet Fabric) (Config) #interface 0/1
(Ethernet Fabric) (Config) (interface 0/1) #routing
(Ethernet Fabric) (Config) (interface 0/1) #ip address 10.10.10.1 255.255.255.0
(Ethernet Fabric) (Config) (interface 0/1) #ip ospf area 0.0.0.0
(Ethernet Fabric) (Config) (interface 0/1) #exit

(Ethernet Fabric) (Config) #interface 0/2
(Ethernet Fabric) (Config) (interface 0/2) #routing
(Ethernet Fabric) (Config) (interface 0/2) #ipv6 enable
(Ethernet Fabric) (Config) (interface 0/2) #ipv6 address 2020:2::2/64
(Ethernet Fabric) (Config) (interface 0/2) #ipv6 ospf
(Ethernet Fabric) (Config) (interface 0/2) #ipv6 ospf network point-to-point
(Ethernet Fabric) (Config) (interface 0/2) #exit

(Ethernet Fabric) (Config) #interface tunnel 0
(Ethernet Fabric) (Config) (interface tunnel 0) #ipv6 address 2001::2/64
(Ethernet Fabric) (Config) (interface tunnel 0) #tunnel mode ipv6ip
(Ethernet Fabric) (Config) (interface tunnel 0) #tunnel source 10.10.10.1
(Ethernet Fabric) (Config) (interface tunnel 0) #tunnel destination 20.20.20.1
(Ethernet Fabric) (Config) (interface tunnel 0) #ipv6 ospf
(Ethernet Fabric) (Config) (interface tunnel 0) #ipv6 ospf network point-to-point
(Ethernet Fabric) (Config) (interface tunnel 0) #exit

(Ethernet Fabric) (Config) #interface loopback 0
(Ethernet Fabric) (Config) (interface loopback 0) #ip address 2.2.2.2 255.255.255.0
(Ethernet Fabric) (Config) (interface loopback 0) #exit
(Ethernet Fabric) (Config) #exit
```

5.3 DHCPv6

DHCP is generally used between clients (e.g., hosts) and servers (e.g., routers) for the purpose of assigning IP addresses, gateways, and other networking definitions such as DNS, NTP, and/or SIP parameters. However, IPv6 natively provides for autoconfiguration of IP addresses through IPv6 Neighbor Discovery Protocol (NDP) and through the use of Router Advertisement messages. Thus, the role of DHCPv6 within the network is different than that of DHCPv4 in that it is less relied upon for IP address assignment.

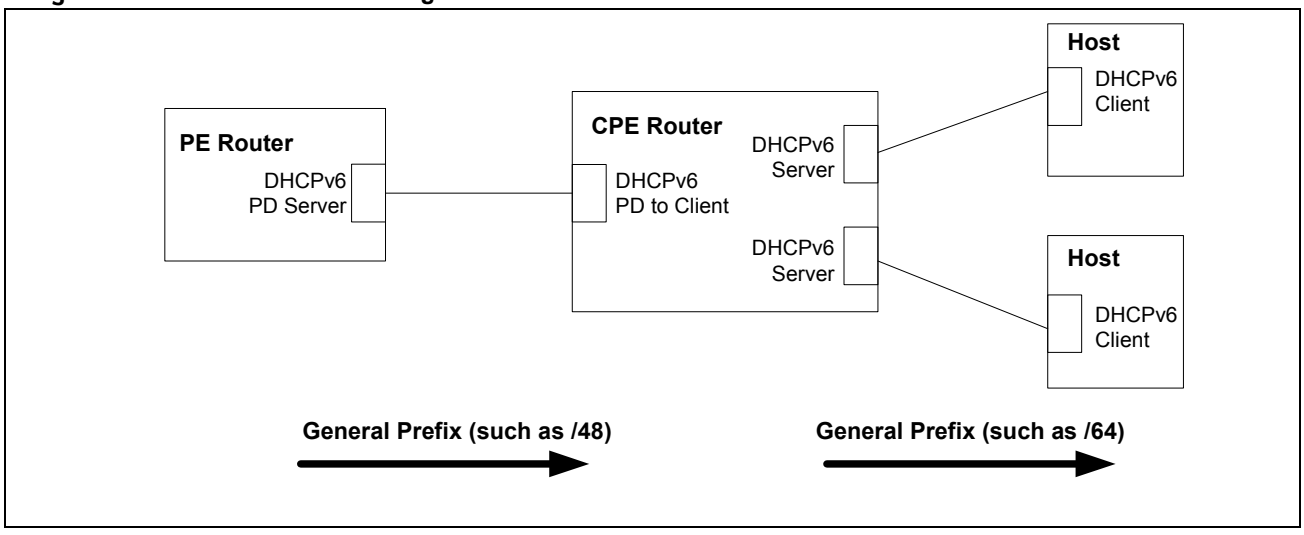
DHCPv6 server and client interactions are described by RFC 3315 [6]. There are many similarities between DHCPv6 and DHCPv4 interactions and options, but the messages and option definitions are sufficiently different such that there is no DHCPv4 to DHCPv6 migration or interoperability.

DHCPv6 incorporates the notion of the "stateless" server, where DHCPv6 is not used for IP address assignment to a client; rather, it only provides other networking information such as DNS, NTP, and/or SIP information. The stateless server behavior is described by RFC 3736 [7], which simply contains descriptions of the portions of RFC 3315 that are necessary for "stateless" server behavior. In order for a router to drive a DHCPv6 client to utilize stateless DHCPv6, the "other stateful configuration" option must be configured for neighbor discovery on the corresponding IPv6 router interface. This, in turn, causes DHCPv6 clients to send the DHCPv6 "Information Request" message in response. A DHCPv6 server then responds by providing only networking definitions such as DNS domain name and server definitions, NTP server definitions, and/or SIP definitions.

RFC 3315 also describes DHCPv6 Relay Agent interactions, which are very much like DHCPv4 Relay Agents. Additionally, there is a DHCPv6 Relay Agent Option Internet draft [9], which employs very similar capabilities as those described by DHCPv4 Relay Agent Option in RFC 2132.

With the larger address space inherent to IPv6, addresses within a network can be allocated more effectively in a hierarchical fashion. DHCPv6 introduces the notion of "prefix delegation" as described in RFC 3633 [8] as a way for routers to centralize and delegate IP address assignment. The following diagram depicts a typical network scenario where prefix delegation is used.

Figure 5-2: DHCPv6 Prefix Delegation Scenario



In Figure 5-2, the PE router acts as Prefix Delegation server and defines one or more "general" prefixes to delegate to a CPE router acting as a Prefix Delegation client. The CPE router then can then allocate more specific addresses within the given general prefix range to assign to its local router interfaces. The CPE router can in turn use the given general prefix in allocating and assigning addresses to host machines that may be utilizing IPv6 auto-address configuration or acting as DHCPv6 clients.

5.3.1 CLI Examples

Just as with DHCPv4 service in FASTPATH, DHCPv6 is disabled by default and can be enabled using the following CLI configuration:

Enable DHCPv6:

```
(Ethernet Fabric) #config
(Ethernet Fabric) (Config) #Service dhcpv6
(Ethernet Fabric) (Config) #exit
```

DHCPv6 pool configuration:

```
(Ethernet Fabric) # config
(Ethernet Fabric) (Config) #ipv6 dhcp pool testpool
(Ethernet Fabric) (Config-dhcp6s-pool) #domain-name lvl7.com
(Ethernet Fabric) (Config-dhcp6s-pool) #dns-server 2001::1
(Ethernet Fabric) (Config-dhcp6s-pool) #exit
(Ethernet Fabric) (Config) #exit
```

Per-interface DHCPv6 configuration:

```
(Ethernet Fabric) #config
(Ethernet Fabric) (Config) #Interface 0/1
(Ethernet Fabric) (Config) (interface 0/1) #ipv6 dhcp server testpool preference 10
(Ethernet Fabric) (Config) (interface 0/1) #exit
(Ethernet Fabric) (Config) #exit
```

6. Quality of Service

This chapter includes the following sections:

- Class of Service Queuing
- Differentiated Services

6.1 Class of Service Queuing

The Class of Service (CoS) feature lets you give preferential treatment to certain types of traffic over others. To set up this preferential treatment, you can configure the ingress ports, the egress ports, and individual queues on the egress ports to provide customization that suits your environment.

The level of service is determined by the egress port queue to which the traffic is assigned. When traffic is queued for transmission, the rate at which it is serviced depends on how the queue is configured and possibly the amount of traffic present in other queues for that port.

Some traffic is classified for service (i.e., packet marking) before it arrives at the switch. If you decide to use these classifications, you can map this traffic to egress queues by setting up a CoS Mapping table.

Each ingress port on the switch has a default priority value (set by configuring VLAN Port Priority in the Switching sub-menu) that determines the egress queue its traffic gets forwarded to. Packets that arrive without a priority designation, or packets from ports you've identified as "untrusted," get forwarded according to this default.

6.1.1 Ingress Port Configuration

6.1.1.1 Trusted and Untrusted Ports/CoS Mapping Table

The first task for ingress port configuration is to specify whether traffic arriving on a given port is "trusted" or "untrusted."

A trusted port means that the system will accept at face value a priority designation within arriving packets. You can configure the system to trust priority designations based on one of the following fields in the packet header:

- 802.1 Priority: values 0-7
- IP DSCP: values 0-63
- IP Precedence: values 0-7

You can also configure an ingress port as untrusted, where the system ignores priority designations of incoming packets and sends the packet to a queue based on the ingress port's default priority.

6.1.1.2 CoS Mapping Table for Trusted Ports

Mapping is from the designated field values on trusted ports' incoming packets to a traffic class priority (actually a CoS traffic queue). The trusted port field-to-traffic class configuration entries form the Mapping Table the switch uses to direct ingress packets from trusted ports to egress queues.

6.1.2 Egress Port Configuration—Traffic Shaping

For slot/port interfaces, you can specify the shaping rate for the port, which is an upper limit of the transmission bandwidth used, specified as a percentage of the maximum link speed.

6.1.3 Queue configuration

For each queue, you can specify:

- Minimum bandwidth guarantee
- Scheduler type – strict/weighted: Strict priority scheduling gives an absolute priority, with highest priority queues always sent first, and lowest priority queues always sent last. Weighted scheduling requires a specification of priority for each queue relative to the other queues, based on their minimum bandwidth values.
- Queue management – tail drop

6.1.4 Queue Management Type

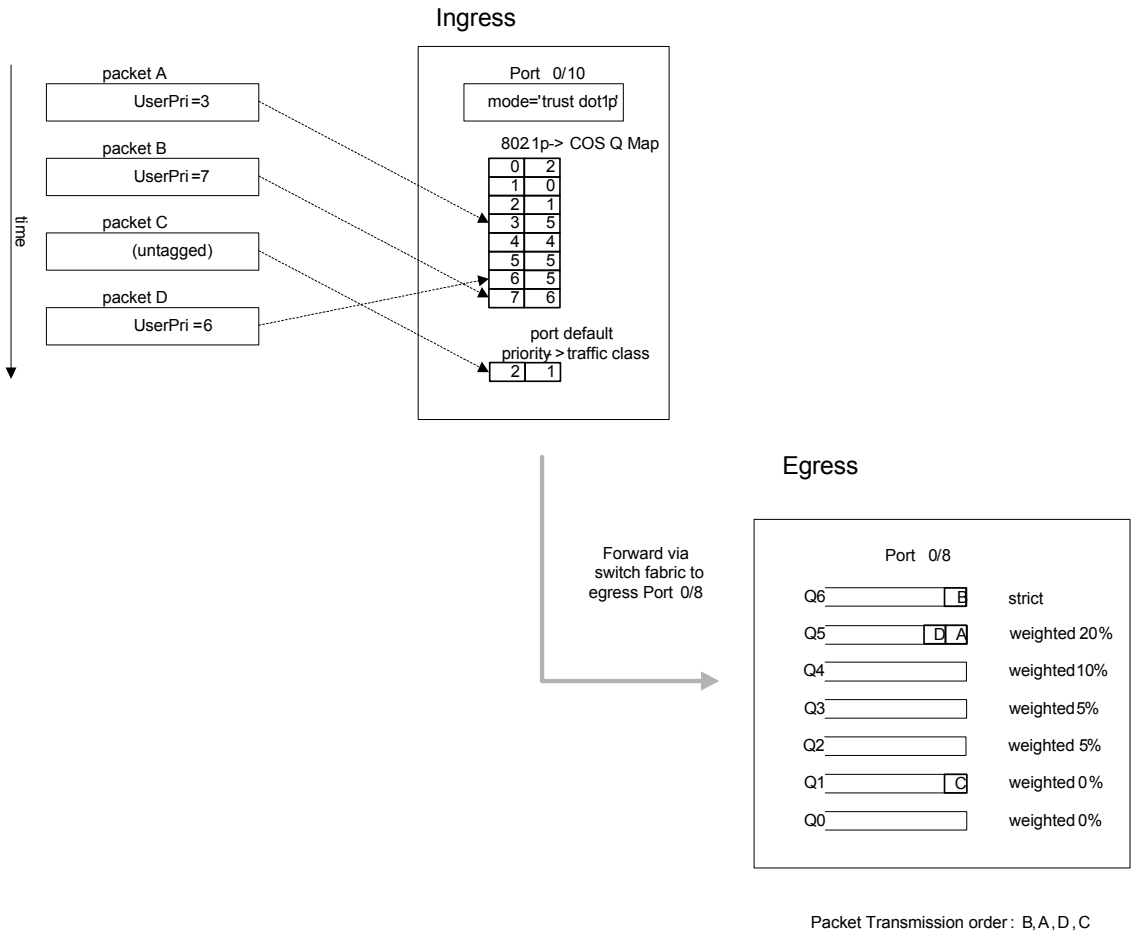
FASTPATH supports the tail drop method of queue management. This means that any packet forwarded to a full queue is dropped regardless of its importance.

6.1.5 CLI Examples

Figure 6-1 on the next page illustrates the network operation as it relates to CoS mapping and queue configuration.

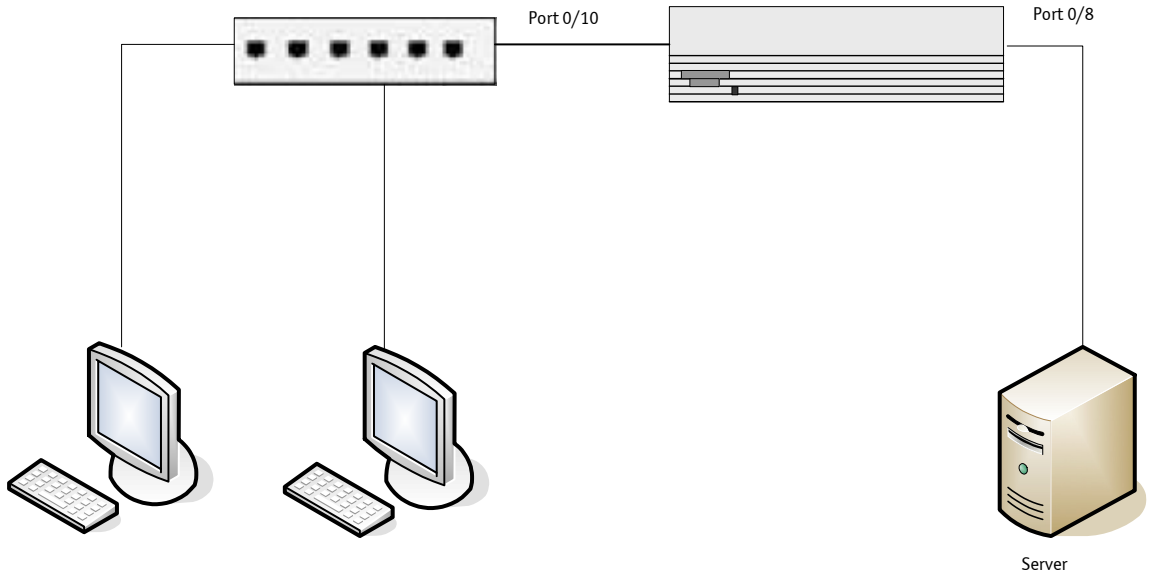
Four packets arrive at the ingress port 0/10 in the order A, B, C, and D. You've configured port 0/10 to trust the 802.1p field of the packet, which serves to direct packets A, B, and D to their respective queues on the egress port. These three packets utilize port 0/10's 802.1p to COS Mapping Table. In this case, the 802.1p user priority 3 was set up to send the packet to queue 5 instead of the default queue 3. Since packet C does not contain a VLAN tag, the 802.1p user priority does not exist, so Port 0/10 relies on its default port priority (2) to direct packet C to egress queue 1.

Figure 6-1: CoS Mapping and Queue Configuration



Continuing this example, you configured the egress Port 0/8 for strict priority on queue 6, and a set a weighted scheduling scheme for queues 5-0. Assuming queue 5 has a higher weighting than queue 1 (relative weight values shown as a percentage, with 0% indicating the bandwidth is not guaranteed), the queue service order is 6 followed by 5 followed by 1. Assuming each queue unloads all packets shown in the diagram, the packet transmission order as seen on the network leading out of Port 0/8 is B, A, D, C. Thus, packet B, with its higher user precedence than the others, is able to work its way through the device with minimal delay and is transmitted ahead of the other packets at the egress port.

Figure 6-2: CoS Configuration Example System Diagram



You will configure the ingress interface uniquely for all cos-queue and VLAN parameters.

```
(Ethernet Fabric) #config
   interface  0/10
      classofservice trust dot1p
      classofservice dot1p-mapping 6 3
      vlan priority 2
      exit

   interface  0/8
      cos-queue min-bandwidth 0 0 5 5 10 20 40
      cos-queue strict 6
      exit
   exit
```

You can also set traffic shaping parameters for the interface. If you wish to shape the egress interface for a sustained maximum data rate of 80 Mbps (assuming a 100Mbps link speed), you would add a simple configuration line expressing the shaping rate as a percentage of link speed.

```
(Ethernet Fabric) #config
   interface  0/8
      traffic-shape 80
      exit
   exit
```

6.2 Differentiated Services

Differentiated Services (DiffServ) is one technique for implementing Quality of Service (QoS) policies. Using DiffServ in your network allows you to directly configure the relevant parameters on the switches and routers rather than using a resource reservation protocol.This section explains how to configure FASTPATH software to identify which traffic class a packet belongs to, and how it should be handled to provide the desired quality of service. As implemented in FASTPATH software, DiffServ allows you to control what traffic is accepted and what traffic is discarded.

Traffic to be processed by the DiffServ feature requires an IP header if the system uses IP Precedence or IP DSCP marking.

How you configure DiffServ support in FASTPATH software varies depending on the role of the switch in your network:

- **Edge device**: An edge device handles ingress traffic, flowing towards the core of the network, and egress traffic, flowing away from the core. An edge device segregates inbound traffic into a small set of traffic classes, and is responsible for determining a packet's classification. Classification is primarily based on the contents of the Layer 3 and Layer 4 headers, and is recorded in the Differentiated Services Code Point (DSCP) added to a packet's IP header.
- **Interior node**: A switch in the core of the network is responsible for forwarding packets, rather than for classifying them. It decodes the DSCP in an incoming packet, and provides buffering and forwarding services using the appropriate queue management algorithms.

Before configuring DiffServ on a particular FASTPATH switch, you must determine the QoS requirements for the network as a whole. The requirements are expressed in terms of rules, which are used to classify inbound traffic on a particular interface. FASTPATH does not support DiffServ in the outbound direction.

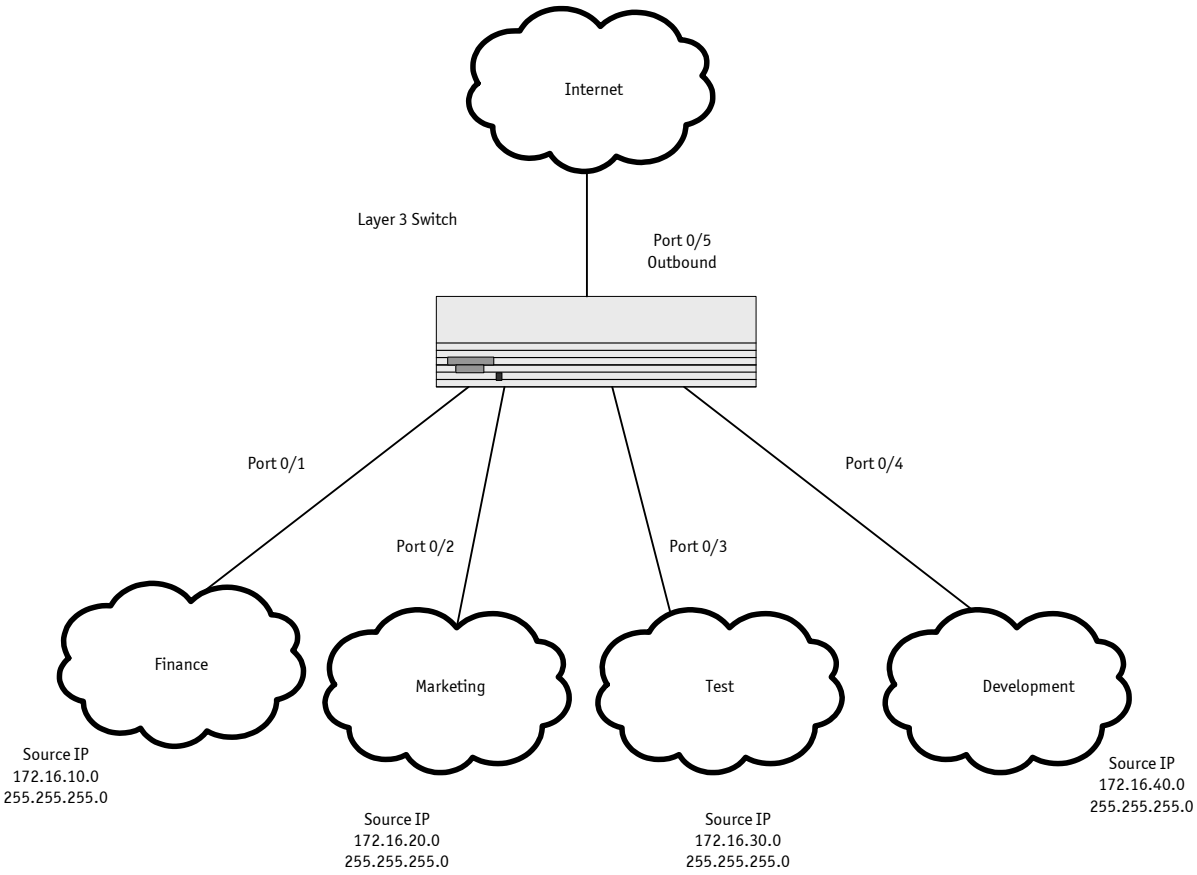
During configuration, you define DiffServ rules in terms of classes, policies and services:

- **Class**: A class consists of a set of rules that identify which packets belong to the class. Inbound traffic is separated into traffic classes based on Layer 2, Layer 3, and Layer 4 header data. One class type is supported, **All,** which specifies that every match criterion defined for the class must be true for a match to occur.
- **Policy**: Defines the QoS attributes for one or more traffic classes. An example of an attribute is the ability to mark a packet at ingress. FASTPATH supports the ability to assign traffic classes to output CoS queues, and to mirror incoming packets in a traffic stream to a specific egress interface (physical port or LAG).
  FASTPATH software supports the **Traffic Conditioning Policy** type which is associated with an inbound traffic class and specifies the actions to be performed on packets meeting the class rules:
  - Marking the packet with a given DSCP, IP precedence, or CoS
  - Policing packets by dropping or re-marking those that exceed the class's assigned data rate
  - Counting the traffic within the class
- **Service –** Assigns a policy to an interface for inbound traffic.

6.2.1 CLI Example

This example shows how a network administrator can provide equal access to the Internet (or other external network) to different departments within a company. Each of four departments has its own Class B subnet that is allocated 25% of the available bandwidth on the port accessing the Internet.

Figure 6-3: DiffServ Internet Access Example Network Diagram



6.2.1.1 Example #1: DiffServ Inbound Configuration

Ensure DiffServ operation is enabled for the switch.

```
(Ethernet Fabric) #config
  diffserv
```

Create a DiffServ class of type “all” for each of the departments, and name them. Define the match criteria—Source IP address—for the new classes.

```
class-map match-all finance_dept
   match srcip 172.16.10.0 255.255.255.0
   exit

class-map match-all marketing_dept
   match srcip 172.16.20.0 255.255.255.0
   exit
```

```
class-map match-all test_dept
   match srcip 172.16.30.0 255.255.255.0
   exit

class-map match-all development_dept
   match srcip 172.16.40.0 255.255.255.0
   exit
```

Create a DiffServ policy for inbound traffic named 'internet_access', adding the previously created department classes as instances within this policy. This policy uses the assign-queue attribute to put each department's traffic on a different egress queue. This is how the DiffServ inbound policy connects to the CoS queue settings established below.

```
policy-map internet_access in
   class finance_dept
      assign-queue 1
      exit
   class marketing_dept
      assign-queue 2
      exit
   class test_dept
      assign-queue 3
      exit
   class development_dept
      assign-queue 4
      exit
   exit
```

Attach the defined policy to interfaces 0/1 through 0/4 in the inbound direction

```
interface 0/1
   service-policy in internet_access
   exit
interface 0/2
   service-policy in internet_access
   exit
interface 0/3
   service-policy in internet_access
   exit
interface 0/4
   service-policy in internet_access
   exit
```

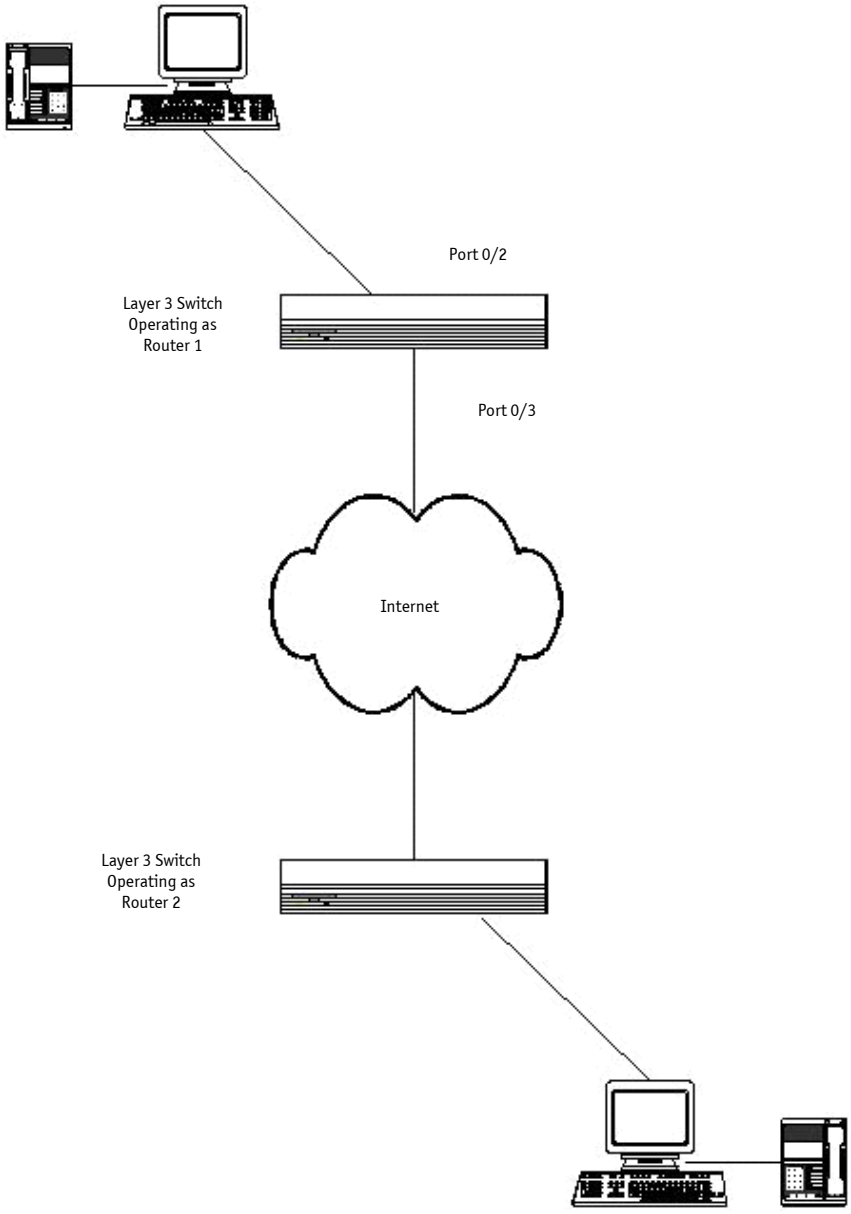
Set the CoS queue configuration for the (presumed) egress interface 0/5 such that each of queues 1, 2, 3 and 4 get a minimum guaranteed bandwidth of 25%. All queues for this interface use weighted round robin scheduling by default. The DiffServ inbound policy designates that these queues are to be used for the departmental traffic through the assign-queue attribute. It is presumed that the switch will forward this traffic to interface 0/5 based on a normal destination address lookup for internet traffic.

```
interface 0/5
   cos-queue min-bandwidth 0 25 25 25 25 0 0
   exit
exit
```

6.2.2 DiffServ for VoIP Configuration Example

One of the most valuable uses of DiffServ is to support Voice over IP (VoIP). VoIP traffic is inherently time-sensitive: for a network to provide acceptable service, a guaranteed transmission rate is vital. This example shows one way to provide the necessary quality of service: how to set up a class for UDP traffic, have that traffic marked on the inbound side, and then expedite the traffic on the outbound side. The configuration script is for Router 1 in the accompanying diagram: a similar script should be applied to Router 2.

Figure 6-4: DiffServ VoIP Example Network Diagram



6.2.2.1 Example #2: Configuring DiffServ VoIP Support

Enter Global Config mode. Set queue 5 on all ports to use strict priority mode. This queue shall be used for all VoIP packets. Activate DiffServ for the switch.

```
(Ethernet Fabric) #config
   cos-queue strict 5
   diffserv
```

Create a DiffServ classifier named 'class_voip' and define a single match criterion to detect UDP packets. The class type "match-all" indicates that all match criteria defined for the class must be satisfied in order for a packet to be considered a match.

```
   class-map match-all class_voip
      match protocol udp
      exit
```

Create a second DiffServ classifier named 'class_ef' and define a single match criterion to detect a DiffServ code point (DSCP) of 'EF' (expedited forwarding). This handles incoming traffic that was previously marked as expedited elsewhere in the network.

```
   class-map match-all class_ef
      match ip dscp ef
      exit
```

Create a DiffServ policy for inbound traffic named 'pol_voip', then add the previously created classes 'class_ef' and 'class_voip' as instances within this policy.

This policy handles incoming packets already marked with a DSCP value of 'EF' (per 'class_ef' definition), or marks UDP packets per the 'class_voip' definition) with a DSCP value of 'EF'. In each case, the matching packets are assigned internally to use queue 5 of the egress port to which they are forwarded.

```
   policy-map pol_voip in
      class class_ef
      assign-queue 5
      exit
   class class_voip
      mark ip-dscp ef
      assign-queue 5
      exit
   exit
```

Attach the defined policy to an inbound service interface.

```
   interface 0/2
      service-policy in pol_voip
      exit
   exit
```

7. Multicast

7.1 Overview

IP Multicasting enables a network host (or multiple hosts) to send an IP datagram to multiple destinations simultaneously. The initiating host sends each multicast datagram only once to a destination multicast group address, and multicast routers forward the datagram only to hosts who are members of the multicast group. Multicast enables efficient use of network bandwidth, as each multicast datagram needs to be transmitted only once on each network link, regardless of the number of destination hosts. Multicasting contrasts with IP unicasting, which sends a separate datagram to each recipient host.

Hosts must have a way to identify their interest in joining any particular multicast group, and routers must have a way to collect and maintain group memberships: these functions are handled by the IGMP protocol in IPv4. In IPv6, multicast routers use the Multicast Listener Discover (MLD) protocol to maintain group membership information.

Multicast routers must also be able to construct a multicast distribution tree that enables forwarding multicast datagrams only on the links that are required to reach a destination group member. Protocols such as DVMRP, and PIM handle this function.

This chapter describes the following multicast protocols:

- IGMP Configuration
- IGMP Proxy
- MLD
- DVMRP
- PIM

7.2 IGMP Configuration

The Internet Group Management Protocol (IGMP) is used by IPv4 hosts to send requests to join (or leave) multicast groups so that they receive (or discontinue receiving) packets sent to those groups.

In IPv4 multicast networks, multicast routers are configured with IGMP so that they can receive join and leave request from directly-connected hosts. They use this information to build a multicast forwarding table.

IPv6 multicast routers use the MLD protocol to perform the functions that IGMP performs in IPv4 networks.

7.2.1 CLI Example

The following example configures IGMP on a FASTPATH router. IP routing, IP multicast, and IGMP are globally enabled on the router. Then, IGMP is configured on the selected interface(s).

```
(Ethernet Fabric) #configure
(Ethernet Fabric) (Config) #ip routing
(Ethernet Fabric) (Config) #ip multicast
(Ethernet Fabric) (Config) #ip igmp
(Ethernet Fabric) (Config) #interface 0/1
(Ethernet Fabric) (Config) (interface 0/1) #routing
(Ethernet Fabric) (Config) (interface 0/1) #ip address 3.3.3.1 255.255.255.0
(Ethernet Fabric) (Config) (interface 0/1) #ip igmp
(Ethernet Fabric) (Config) (interface 0/1) #exit
(Ethernet Fabric) (Config) #exit
```

A multicast router must also have a way to determine how to efficiently forward multicast packets. The information gathered by IGMP is provided to a multicast routing protocol (DVMRP, PIM-DM, and PIM-SM) configured on the router to ensure that multicast packets are delivered to all networks where there are interested receivers. Refer to those sections for configuration instructions.

7.3 IGMP Proxy

IGMP proxy enables a multicast router to learn multicast group membership information and forward multicast packets based upon the group membership information. The IGMP Proxy is capable of functioning only in certain topologies that do not require Multicast Routing Protocols (DVMRP, PIM-DM, and PIM-SM) and have a tree-like topology, as there is no support for features like reverse path forwarding (RPF) to correct packet route loops.

The proxy contains many downstream interfaces and a unique upstream interface explicitly configured. It performs the host side of the IGMP protocol on its upstream interface and the router side of the IGMP protocol on its downstream interfaces.

The IGMP proxy offers a mechanism for multicast forwarding based only on IGMP membership information. The router must decide about forwarding packets on each of its interfaces based on the IGMP membership information. The proxy creates the forwarding entries based on the membership information and adds it to the multicast forwarding cache (MFC) in order not to make the forwarding decision for subsequent multicast packets with same combination of source and group.

7.3.1 CLI Examples

The CLI component of FASTPATH allows the end users to configure the network device and to view device settings and statistics using a serial interface or telnet session.

7.3.1.1 Example #1: Configuring IGMP Proxy on the Router

This command enables the IGMP Proxy on the router. To enable IGMP Proxy on the router no multicast routing protocol should be enabled and also multicast forwarding must be enabled on the router. Use these commands from the Interface mode:

```
(Ethernet Fabric) #configure
(Ethernet Fabric) (Config) #ip routing
(Ethernet Fabric) (Config) #ip multicast
(Ethernet Fabric) (Config) #ip igmp
(Ethernet Fabric) (Config) #interface 0/15
(Ethernet Fabric) (Config) (interface 0/15) #ip igmp-proxy
```

Additional configuration options are available for the igmp-proxy command:

```
<cr>                          Press Enter to execute the command.
reset-status                  Reset All the proxy interface status parameters.
unsolicited-report-intervalConfigure IGMP Proxy unsolicited report interval.
```

The value of the unsolicited report interval can range from 1 to 260 seconds. The default is 1 second. Use this command from the Interface mode.

7.3.1.2 Example #2: View IGMP Proxy Configuration Data

You can use various commands from Privileged EXEC or User EXEC modes to show IGMP proxy configuration data.

- Use the following command to display a summary of the host interface status parameters. It displays the parameters only when IGMP Proxy is enabled.
  ```
  (Ethernet Fabric) #show ip igmp-proxy

  Interface Index.............................. 0/15
  Admin Mode.................................... Enabled
  Operational Mode.............................. Disabled
  ```
- Use the following command to display interface parameters when IGMP Proxy is enabled:
  ```
  (Ethernet Fabric) #show ip igmp-proxy interface
  ```
- Use this command to display information about multicast groups that IGMP proxy reported. It displays a table of entries with the following as the fields of each column.
  ```
  (Ethernet Fabric) #show ip igmp-proxy groups
  ```
- Use the following command to display information about multicast groups that IGMP proxy reported. It displays a table of entries with the following as the fields of each column:
  ```
  (Ethernet Fabric) #show ip igmp-proxy groups detail
  ```

7.4 MLD

Multicast Listener Discovery (MLD) protocol enables IPv6 routers to discover the presence of *multicast listeners*, the nodes who wish to receive the multicast data packets, on its directly-attached interfaces. On IPv6 multicast routers, MLD replaces the functionality performed by IGMP on IPv4 networks.

MLD discovers which multicast addresses are of interest to its neighboring nodes and provides this information to the active multicast routing protocol that makes decisions on the flow of multicast data packets.

The Multicast router sends General Queries periodically to request multicast address listeners information from systems on an attached network. These queries are used to build and refresh the multicast address listener state on attached networks. Multicast listeners respond to these queries by reporting their multicast addresses listener state and their desired set of sources with Current-State Multicast address Records in the MLD2 Membership Reports. The Multicast router also processes unsolicited Filter-Mode-Change records and Source-List-Change Records from systems that want to indicate interest in receiving or not receiving traffic from particular sources.

The FASTPATH implementation of MLD v2 supports the multicast router portion of the protocol (i.e., not the listener portion). It is backward-compatible with MLD v1.

After configuring MLD, you must also enable a multicast routing protocol, which the router uses make multicast routing decisions based on the information collected through MLD. See PIM-SM or PIM-DM.

7.4.1 CLI Example

The following example configures MLD on a router interface.

First, IPv4 and IPv6 routing are globally enabled. Then, MLD is globally enabled. Finally, MLD router is enabled on a specific interface.

```
(Ethernet Fabric) #configure
(Ethernet Fabric) (Config) #ip routing
(Ethernet Fabric) (Config) #ip multicast
(Ethernet Fabric) (Config) #ipv6 unicast-routing
(Ethernet Fabric) (Config) #ipv6 mld router
(Ethernet Fabric) (Config) #interface 0/1
(Ethernet Fabric) (Config) (interface 0/1) #routing
(Ethernet Fabric) (Config) (interface 0/1) #ipv6 address 2006::3/64
(Ethernet Fabric) (Config) (interface 0/1) #ipv6 enable
(Ethernet Fabric) (Config) (interface 0/1) #ipv6 mld router
(Ethernet Fabric) (Config) (interface 0/1) #exit
(Ethernet Fabric) (Config) #exit
```

7.5 DVMRP

The Distance Vector Multicast Routing Protocol (DVMRP) is one of several multicast routing protocols you can configure on a FASTPATH router (PIM-SM and PIM-DM are the others). Note that only one multicast routing protocol (MRP) can be operational on a router at any time.

DVMRP is an interior gateway protocol; i.e., it is suitable for use within an autonomous system, but not between different autonomous systems.

DVMRP is based on RIP: it forwards multicast datagrams to other routers in the AS and constructs a forwarding table based on information it learns in response. More specifically, it uses this sequence.

- A new multicast packet is forwarded to the entire multicast network, with respect to the time-to-live (TTL) of the packet.
- The TTL restricts the area to be flooded by the message.
- All routers that do not have members on directly-attached subnetworks send back *Prune messages* to the upstream router.
- The branches that transmit a prune message are deleted from the delivery tree.
- The delivery tree which is spanning to all the members in the multicast group, is constructed in the form of a DVMRP forwarding table.

7.5.1 CLI Example

The following example configures two DVMRP interfaces. First, this example configures an OSPF router[1] and globally enables IP routing and IP multicast. IGMP is globally enabled so that this router can manage group membership information for its directly-connected hosts (IGMP may not be required when there are no directly connected hosts). Next, DVMRP is globally enabled. Finally, DVMRP, IGMP, and OSPF are enabled on several interfaces.

```
(Ethernet Fabric) #configure
(Ethernet Fabric) (Config) #router ospf
(Ethernet Fabric) (Config-router) #router-id 3.3.1.1
(Ethernet Fabric) (Config-router) #exit
(Ethernet Fabric) (Config) #ip routing
(Ethernet Fabric) (Config) #ip multicast
(Ethernet Fabric) (Config) #ip igmp
(Ethernet Fabric) (Config) #ip dvmrp
(Ethernet Fabric) (Config) #interface 0/1
(Ethernet Fabric) (Config) (interface 0/1) #routing
(Ethernet Fabric) (Config) (interface 0/1) #ip address 3.3.3.1 255.255.255.0
(Ethernet Fabric) (Config) (interface 0/1) #ip dvmrp
(Ethernet Fabric) (Config) (interface 0/1) #ip igmp
(Ethernet Fabric) (Config) (interface 0/1) #ip ospf area 0
(Ethernet Fabric) (Config) (interface 0/1) #exit
(Ethernet Fabric) (Config) #interface 0/3
(Ethernet Fabric) (Config) (interface 0/3) #routing
(Ethernet Fabric) (Config) (interface 0/3) #ip address 1.1.1.1 255.255.255.0
(Ethernet Fabric) (Config) (interface 0/3) #ip dvmrp
(Ethernet Fabric) (Config) (interface 0/3) #ip igmp
(Ethernet Fabric) (Config) (interface 0/3) #ip ospf area 0
(Ethernet Fabric) (Config) (interface 0/3) #exit
(Ethernet Fabric) (Config) #exit
```

1. OSPF configuration is added as a unicast protocol for illustration purposes; static unicast routing could also be configured.

7.6 PIM

Protocol Independent Multicast (PIM) is a standard multicast routing protocol that provides scalable inter-domain multicast routing across the Internet, independent of the mechanisms provided by any particular unicast routing protocol.

PIM has two types:

- PIM-Dense Mode (PIM-DM)
- PIM-Sparse Mode (PIM-SM)

7.6.1 PIM-SM

PIM-SM is used to efficiently route multicast traffic to multicast groups that may span wide area networks where bandwidth is a constraint.

PIM-SM uses shared trees by default and implements source-based trees for efficiency; it assumes that no hosts want the multicast traffic unless they specifically ask for it. It creates a shared distribution tree centered on a defined "*rendezvous point" (RP)* from which source traffic is relayed to the receivers. Senders first send the multicast data to the RP, which in turn sends the data down the shared tree to the receivers. Shared trees centered on an RP do not necessarily provide the shortest, most optimal path. In such cases, PIM-SM provides a means to switch to more efficient source-specific trees. A data threshold rate is configured to determine when to switch from shared-tree to source-tree.

PIM-SM uses a *Bootstrap Router (BSR)*, which advertises information to other multicast routers about the RP. In a given network, a set of routers can be administratively enabled as candidate bootstrap routers. If it is not apparent which router should be the BSR, the candidates flood the domain with advertisements. The router with the highest priority is elected. If all the priorities are equal, then the candidate with the highest IP address becomes the BSR.

PIM-SM is defined in RFC 4601.

7.6.1.1 Example #1: PIM-SMv4

The following example configures PIM-SM for IPv4 on a router.

First, configure an OSPF[1] router and globally enable IP routing, multicast, IGMP, and PIM-SM. Next, configure a PIM-SM rendezvous point with an IP address and group range. The IP address will serve as an RP for the range of potential multicast groups specified in the group range. Finally, enable routing, IGMP, PIM-SM, and OSPF on one or more interfaces.

```
(Ethernet Fabric) #configure
(Ethernet Fabric) (Config) #router ospf
(Ethernet Fabric) (Config-router) #router-id 3.3.1.1
(Ethernet Fabric) (Config-router) #exit
(Ethernet Fabric) (Config) #ip routing
(Ethernet Fabric) (Config) #ip multicast
(Ethernet Fabric) (Config) #ip igmp
(Ethernet Fabric) (Config) #ip pimsm          [NOTE: This router should be an RP.]
(Ethernet Fabric) (Config) #ip pimsm rp-address 1.1.1.1 224.0.0.0 240.0.0.0
```

1. OSPF configuration is added as a unicast protocol for illustration purposes; static unicast routing could also be configured.

```
(Ethernet Fabric) (Config) #interface 0/1
(Ethernet Fabric) (Config) (interface 0/1) #routing
(Ethernet Fabric) (Config) (interface 0/1) #ip address 3.3.3.1 255.255.255.0
(Ethernet Fabric) (Config) (interface 0/1) #ip pimsm
(Ethernet Fabric) (Config) (interface 0/1) #ip igmp
(Ethernet Fabric) (Config) (interface 0/1) #ip ospf area 0
(Ethernet Fabric) (Config) (interface 0/1) #exit
(Ethernet Fabric) (Config) #interface 0/3
(Ethernet Fabric) (Config) (interface 0/3) #routing
(Ethernet Fabric) (Config) (interface 0/3) #ip address 1.1.1.1 255.255.255.0
(Ethernet Fabric) (Config) (interface 0/3) #ip pimsm
(Ethernet Fabric) (Config) (interface 0/3) #ip igmp
(Ethernet Fabric) (Config) (interface 0/3) #ip ospf area 0
(Ethernet Fabric) (Config) (interface 0/3) #exit
(Ethernet Fabric) (Config) #exit
```

7.6.1.2 Example #2: PIM-SMv6

The following example configures PIM-SM for IPv6 on a router.

First, configure an OSPF[1] router for IPv6 and globally enable IPv4 and IPv6 routing, multicast, MLD, and PIM-SM. Next, configure a PIM-SM rendezvous point with an IP address and a group range. The IP address will serve as an RP for the range of potential multicast groups specified in the group range. Finally, enable routing, IGMP, PIM-SM, and OSPF on one or more interfaces.

```
(Ethernet Fabric) #configure
(Ethernet Fabric) (Config) #ipv6 router ospf
       router-id 3.3.2.2
       exit
(Ethernet Fabric) (Config) #ip routing
(Ethernet Fabric) (Config) #ip multicast
(Ethernet Fabric) (Config) #ipv6 unicast-routing
(Ethernet Fabric) (Config) #ipv6 mld router
(Ethernet Fabric) (Config) #ipv6 pimsm [NOTE: This router should be RP.]
(Ethernet Fabric) (Config) #ipv6 pimsm rp-address 2006::3 fffe::1/64
(Ethernet Fabric) (Config) #interface 0/1
(Ethernet Fabric) (Config) (interface 0/1) #routing
(Ethernet Fabric) (Config) (interface 0/1) #ipv6 address 2006::3/64
(Ethernet Fabric) (Config) (interface 0/1) #ipv6 enable
(Ethernet Fabric) (Config) (interface 0/1) #ipv6 mld router
(Ethernet Fabric) (Config) (interface 0/1) #ipv6 pimsm
(Ethernet Fabric) (Config) (interface 0/1) #ipv6 ospf
(Ethernet Fabric) (Config) (interface 0/1) #exit
(Ethernet Fabric) (Config) #interface 0/5
(Ethernet Fabric) (Config) (interface 0/5) #routing
(Ethernet Fabric) (Config) (interface 0/5) #ipv6 address 2004::3/64
(Ethernet Fabric) (Config) (interface 0/5) #ipv6 enable
(Ethernet Fabric) (Config) (interface 0/5) #ipv6 mld router
(Ethernet Fabric) (Config) (interface 0/5) #ipv6 pimsm
(Ethernet Fabric) (Config) (interface 0/5) #ipv6 ospf
(Ethernet Fabric) (Config) (interface 0/5) #exit
(Ethernet Fabric) (Config) #exit
```

1. OSPF configuration is added as a unicast protocol for illustration purposes; static unicast routing could also be configured.

7.6.2 PIM-DM

PIM-DM protocol is a simple, protocol-independent multicast routing protocol. It uses existing unicast routing table and join/prune/graft mechanism to build a tree. PIM-DM creates source-based shortest-path distribution trees making use of Reverse Path Forwarding (RPF).

PIM-DM cannot be used to build a shared distribution tree, as PIM-SM can. PIM-DM assumes that when a sender starts sending data, all downstream routers and hosts want to receive a multicast datagram. PIM-DM initially floods multicast traffic throughout the network. Routers that do not have any downstream neighbors send back Prune messages that instruct the upstream router to remove that multicast route from its forwarding table. In addition to the Prune messages, PIM-DM makes use of two more messages: Graft and Assert. Graft messages are used whenever a new host wants to join the group. Assert messages are used to shut off duplicate flows onto the same multi-access network.

To minimize the repeated flooding of datagrams and subsequent pruning associated with a particular source-group (S,G) pair, PIM-DM uses a State Refresh message. This message is sent by the router(s) directly connected to the source and is propagated throughout the network. When received by a router on its RPF interface, the State Refresh message causes an existing prune state to be refreshed. State Refresh messages are generated periodically by the router directly attached to the source.

PIM-DM is appropriate for:

- Densely distributed receivers
- A ratio of few senders-to-many receivers (due to frequent flooding)
- High volume of multicast traffic
- Constant stream of traffic

7.6.2.1 Example #1: PIM-DMv4

The following example configures PIM-DM for IPv4 on a router.

First, configure an OSPF[1] router and globally enable IP routing, multicast, IGMP, and PIM-DM. Next, enable routing, IGMP, PIM-DM, and OSPF on one more interfaces.

```
(Ethernet Fabric) #configure
(Ethernet Fabric) (Config) #router ospf
(Ethernet Fabric) (Config-router) #router-id 3.3.1.1
(Ethernet Fabric) (Config-router) #exit
(Ethernet Fabric) (Config) #ip routing
(Ethernet Fabric) (Config) #ip multicast
(Ethernet Fabric) (Config) #ip igmp
(Ethernet Fabric) (Config) #ip pimdm
(Ethernet Fabric) (Config) #interface 0/1
(Ethernet Fabric) (Config) (interface 0/1) #routing
(Ethernet Fabric) (Config) (interface 0/1) #ip address 3.3.3.1 255.255.255.0
(Ethernet Fabric) (Config) (interface 0/1) #ip pimdm
(Ethernet Fabric) (Config) (interface 0/1) #ip igmp
(Ethernet Fabric) (Config) (interface 0/1) #ip ospf area 0
(Ethernet Fabric) (Config) (interface 0/1) #exit
(Ethernet Fabric) (Config) #interface 0/3
(Ethernet Fabric) (Config) (interface 0/3) #routing
(Ethernet Fabric) (Config) (interface 0/3) #ip address 1.1.1.1 255.255.255.0
(Ethernet Fabric) (Config) (interface 0/3) #ip pimdm
(Ethernet Fabric) (Config) (interface 0/3) #ip igmp
(Ethernet Fabric) (Config) (interface 0/3) #ip ospf area 0
(Ethernet Fabric) (Config) (interface 0/3) #exit
(Ethernet Fabric) (Config) #exit
```

1. OSPF configuration is added as a unicast protocol for illustration purposes; static unicast routing could also be configured.

7.6.2.2 Example #2: PIM-DMv6

The following example configures PIM-DM for IPv6 on a router.

First, configure an OSPF[1] router and globally enable IPv4 and IPv6 routing, multicast, MLD, and PIM-DM. Next, enable routing, MLD, PIM-DM, and OSPFv6 on one more interfaces.

```
(Ethernet Fabric) #configure
(Ethernet Fabric) (Config) #ipv6 router ospf
(Ethernet Fabric) (Config) (Config-router) #router-id 3.3.2.2
(Ethernet Fabric) (Config) (Config-router) #exit
(Ethernet Fabric) (Config) #ip routing
(Ethernet Fabric) (Config) #ip multicast
(Ethernet Fabric) (Config) #ipv6 unicast-routing
(Ethernet Fabric) (Config) #ipv6 mld router
(Ethernet Fabric) (Config) #ipv6 pimdm
(Ethernet Fabric) (Config) #interface 0/1
(Ethernet Fabric) (Config) (interface 0/1) #routing
(Ethernet Fabric) (Config) (interface 0/1) #ipv6 address 2006::3/64
(Ethernet Fabric) (Config) (interface 0/1) #ipv6 enable
(Ethernet Fabric) (Config) (interface 0/1) #ipv6 mld router
(Ethernet Fabric) (Config) (interface 0/1) #ipv6 pimdm
(Ethernet Fabric) (Config) (interface 0/1) #ipv6 ospf
(Ethernet Fabric) (Config) (interface 0/1) #exit
(Ethernet Fabric) (Config) #interface 0/5
(Ethernet Fabric) (Config) (interface 0/5) #routing
(Ethernet Fabric) (Config) (interface 0/5) #ipv6 address 2004::3/64
(Ethernet Fabric) (Config) (interface 0/5) #ipv6 enable
(Ethernet Fabric) (Config) (interface 0/5) #ipv6 mld router
(Ethernet Fabric) (Config) (interface 0/5) #ipv6 pimdm
(Ethernet Fabric) (Config) (interface 0/5) #ipv6 ospf
(Ethernet Fabric) (Config) (interface 0/5) #exit
(Ethernet Fabric) (Config) #exit
```

1. OSPF configuration is added as a unicast protocol for illustration purposes; static unicast routing could also be configured.