Part No. 060160-10, Rev. J
April 2006

OmniSwitch 7700/7800
OmniSwitch 8800
Network Configuration Guide

ALCATEL

www.alcatel.com

This user guide documents release 5.4 of the OmniSwitch 7700, OmniSwitch 7800, and OmniSwitch 8800.
The functionality described in this guide is subject to change without notice.

This OmniSwitch product contains components which may be covered by one or more of the following U.S. Patents:

- U.S. Patent No. 6,339,830
- U.S. Patent No. 6,070,243
- U.S. Patent No. 6,061,368
- U.S. Patent No. 5,394,402
- U.S. Patent No. 6,047,024
- U.S. Patent No. 6,314,106
- U.S. Patent No. 6,542,507
- U.S. Patent No. 6,874,090

ALCATEL

26801 West Agoura Road
Calabasas, CA 91301
(818) 880-3500 FAX (818) 880-3505
info@ind.alcatel.com

US Customer Support—(800) 995-2696
International Customer Support—(818) 878-4507
Internet—http://eservice.ind.alcatel.com

Contents

About This Guide

This *OmniSwitch 7700/7800/8800 Network Configuration Guide* describes how to set up and monitor software features that will allow your switch to operate in a live network environment. The software features described in this manual are shipped standard with your OmniSwitch 7700, 7800, or 8800. These features are used when setting up your OmniSwitch in a network of switches and routers.

Supported Platforms

This information in this guide applies to the following products:

- OmniSwitch 7700
- OmniSwitch 7800
- OmniSwitch 8800

The OmniSwitch 7700 includes 10 slots for high performance 10/100 Ethernet and Gigabit Ethernet Network Interface (NI) modules. The OmniSwitch 7800 includes 18 slots for high performance 10/100 Ethernet and Gigabit Ethernet NI modules. The OmniSwitch 8800 includes 18 slots for high performance 10/100 Ethernet and Gigabit Ethernet NI modules.

Unsupported Platforms

The information in this guide does not apply to the following products:

- OmniSwitch (original version with no numeric model name)
- OmniSwitch 6624
- OmniSwitch 6648
- OmniSwitch 6600-U24
- OmniSwitch 6600-P24
- OmniSwitch 6602-24
- OmniSwitch 6602-48
- OmniSwitch 6800-24
- OmniSwitch 6800-48
- OmniSwitch 6800-U24
- OmniSwitch 6800-24L
- OmniSwitch 6800-48L

- OmniSwitch 6850
- OmniSwitch 9700
- Omni Switch/Router
- OmniStack
- OmniAccess

Who Should Read this Manual?

The audience for this user guide is network administrators and IT support personnel who need to configure, maintain, and monitor switches and routers in a live network. However, anyone wishing to gain knowledge on how fundamental software features are implemented in the OmniSwitch 7700, 7800, or 8800 will benefit from the material in this configuration guide.

When Should I Read this Manual?

Read this guide as soon as you are ready to integrate your OmniSwitch into your network of switches and routers. You should already be familiar with the basics of managing a single OmniSwitch as described in the *OmniSwitch 7700/7800/8800 Switch Management Guide*.

The topics and procedures in this manual assume an understanding of the OmniSwitch directory structure and basic switch administration commands and procedures. This manual will help you set up your switches to communicate with other switches in the network. The topics in this guide include VLANs, network routing protocols, authentication, and Quality of Service (QoS)—features that are typically deployed in a multi-switch environment.

What is in this Manual?

This configuration guide includes information about configuring the following features:

- VLANs, VLAN router ports, mobile ports, and VLAN rules.
- Basic Layer 2 functions, such as Ethernet port parameters, source learning, Spanning Tree, and Alcatel interswitch protocols (AMAP and GMAP).
- Advanced Layer 2 functions, such as 802.1Q tagging, Link Aggregation, IP Multicast Switching, and Server Load Balancing.
- Basic routing protocols and functions, such as static IP routes, RIP, DHCP Relay, Virtual Router Redundancy Protocol (VRRP), and IPX.
- Security features, such as switch access control, Authenticated VLANs (AVLANs), authentication servers, and policy management.
- Quality of Service (QoS) and Access Control Lists (ACLs) features, such as policy rules for prioritizing and filtering traffic, remapping packet headers, and network address translation.
- Diagnostic tools, such as RMON, port mirroring, and switch logging.

What is Not in this Manual?

The configuration procedures in this manual use Command Line Interface (CLI) commands in all examples. CLI commands are text-based commands used to manage the switch through serial (console port) connections or via Telnet sessions. Procedures for other switch management methods, such as web-based (WebView or OmniVista) or SNMP, are outside the scope of this guide.

For information on WebView and SNMP switch management methods consult the *OmniSwitch 7700/7800/8800 Switch Management Guide.* Information on using WebView and OmniVista can be found in the context-sensitive on-line help available with those network management applications.

This guide provides overview material on software features, how-to procedures, and application examples that will enable you to begin configuring your OmniSwitch. It is not intended as a comprehensive reference to all CLI commands available in the OmniSwitch. For such a reference to all OmniSwitch CLI commands, consult the *OmniSwitch CLI Reference Guide*.

How is the Information Organized?

Chapters in this guide are broken down by software feature. The titles of each chapter include protocol or features names (e.g., 802.1Q, RIP, server load balancing) with which most network professionals will be familiar.

Each software feature chapter includes sections that will satisfy the information requirements of casual readers, rushed readers, serious detail-oriented readers, advanced users, and beginning users.

Quick Information. Most chapters include a *specifications table* that lists RFCs and IEEE specifications supported by the software feature. In addition, this table includes other pertinent information such as minimum and maximum values and sub-feature support. Most chapters also include a *defaults table* that lists the default values for important parameters along with the CLI command used to configure the parameter. Many chapters include a *Quick Steps* section, which is a procedure covering the basic steps required to get a software feature up and running.

In-Depth Information. All chapters include *overview sections* on the software feature as well as on selected topics of that software feature. *Topical sections* may often lead into *procedure sections* that describe how to configure the feature just described. Serious readers and advanced users will also find the many *application examples*, located near the end of chapters, helpful. Application examples include diagrams of real networks and then provide solutions using the CLI to configure a particular feature, or more than one feature, within the illustrated network.

Documentation Roadmap

The OmniSwitch user documentation suite was designed to supply you with information at several critical junctures of the configuration process. The following section outlines a roadmap of the manuals that will help you at each stage of the configuration process. Under each stage, we point you to the manual or manuals that will be most helpful to you.

Stage 1: Using the Switch for the First Time

Pertinent Documentation: *OmniSwitch 7700/7800 Getting Started Guide*
OmniSwitch 8800 Getting Started Guide
Release Notes

A hard-copy *OmniSwitch 7700/7800 Getting Started Guide* is included with OmniSwitch 7700/7800 switches and a hard-copy *OmniSwitch 8800 Getting Started Guide* is included with OmniSwitch 8800 switches; these guides provide all the information you need to get your switch up and running the first time. These guides provide information on unpacking the switch, rack mounting the switch, installing NI modules, unlocking access control, setting the switch's IP address, and setting up a password. They also include succinct overview information on fundamental aspects of the switch, such as hardware LEDs, the software directory structure, CLI conventions, and web-based management.

At this time you should also familiarize yourself with the Release Notes that accompanied your switch. This document includes important information on feature limitations that are not included in other user guides.

Stage 2: Gaining Familiarity with Basic Switch Functions

Pertinent Documentation: *OmniSwitch 7700/7800 Hardware Users Guide*
OmniSwitch 8800 Hardware Users Guide
OmniSwitch 7700/7800/8800 Switch Management Guide

Once you have your switch up and running, you will want to begin investigating basic aspects of its hard ware and software. Information about OmniSwitch 7700/7800 hardware is provided in the *OmniSwitch 7700/7800 Hardware Users Guide*. Information about OmniSwitch 8800 hardware is provided in the *OmniSwitch 8800 Hardware Users Guide*. These guides provide specifications, illustrations, and descriptions of all hardware components—chassis, power supplies, Chassis Management Modules (CMMs), Network Interface (NI) modules, and cooling fans. They also include steps for common procedures, such as removing and installing switch components.

The *OmniSwitch 7700/7800/8800 Switch Management Guide* is the primary user guide for the basic software features on a single switch. This guide contains information on the switch directory structure, basic file and directory utilities, switch access security, SNMP, and web-based management. It is recommended that you read this guide before connecting your switch to the network.

Stage 3: Integrating the Switch Into a Network

Pertinent Documentation: *OmniSwitch 7700/7800/8800 Network Configuration Guide*
OmniSwitch 7700/7800/8800 Advanced Routing Configuration Guide

When you are ready to connect your switch to the network, you will need to learn how the OmniSwitch implements fundamental software features, such as 802.1Q, VLANs, Spanning Tree, and network routing protocols. The *OmniSwitch 7700/7800/8800 Network Configuration Guide* contains overview information, procedures, and examples on how standard networking technologies are configured in the OmniSwitch 7700, 7800, and 8800.

The *OmniSwitch 7700/7800/8800 Advanced Routing Configuration Guide* includes configuration information for networks using advanced routing technologies, such as OSPF, BGP, and multicast routing protocols (DVMRP and PIM-SM).

Anytime

The *OmniSwitch CLI Reference Guide* contains comprehensive information on all CLI commands supported by the switch. This guide includes syntax, default, usage, example, related CLI command, and CLI-to-MIB variable mapping information for all CLI commands supported by the switch. This guide can be consulted anytime during the configuration process to find detailed and specific information on each CLI command.

Related Documentation

The following are the titles and descriptions of all the OmniSwitch 7700/7800/8800 user manuals:

- *OmniSwitch 7700/7800 Getting Started Guide*

  Describes the hardware and software procedures for getting an OmniSwitch 7700/7800 up and running. Also provides information on fundamental aspects of OmniSwitch software architecture.

- *OmniSwitch 8800 Getting Started Guide*

  Describes the hardware and software procedures for getting an OmniSwitch 8800 up and running. Also provides information on fundamental aspects of OmniSwitch software architecture.

- *OmniSwitch 7700/7800 Hardware Users Guide*

  Complete technical specifications and procedures for all OmniSwitch 7700/7800 chassis, power supplies, Chassis Management Modules (CMMs), fans, and Network Interface (NI) modules.

- *OmniSwitch 8800 Hardware Users Guide*

  Complete technical specifications and procedures for all OmniSwitch 8800 chassis, power supplies, Chassis Management Modules (CMMs), Switch Fabric Modules (SFMs), fans, and Network Interface (NI) modules.

- *OmniSwitch CLI Reference Guide*

  Complete reference to all CLI commands supported on the OmniSwitch 6624, 6648, 7700, 7800, and 8800. Includes syntax definitions, default values, examples, usage guidelines and CLI-to-MIB variable mappings.

- *OmniSwitch 7700/7800/8800 Switch Management Guide*

  Includes procedures for readying an individual switch for integration into a network. Topics include the software directory architecture, image rollback protections, authenticated switch access, managing switch files, system configuration, using SNMP, and using web management software (WebView).

- *OmniSwitch 7700/7800/8800 Network Configuration Guide*

  Includes network configuration procedures and descriptive information on all the major software features and protocols included in the base software package. Chapters cover Layer 2 information (Ethernet and VLAN configuration), Layer 3 information (routing protocols, such as RIP and IPX), security options (authenticated VLANs), Quality of Service (QoS), link aggregation, and server load balancing.

- *OmniSwitch 7700/7800/8800 Advanced Routing Configuration Guide*

  Includes network configuration procedures and descriptive information on all the software features and protocols included in the advanced routing software package. Chapters cover multicast routing (DVMRP and PIM-SM), OSPF, and BGP.

- Technical Tips, Field Notices

  Includes information published by Alcatel's Customer Support group.

- *Release Notes*

  Includes critical Open Problem Reports, feature exceptions, and other important information on the features supported in the current release and any limitations to their support.

User Manuals Web Site

All related user guides for the OmniSwitch 7700, 7800, and 8800 can be found on our web site at http://www.alcatel.com/enterprise/en/resource_library/user_manuals.html

All documentation on the User Manual web site is in PDF format and requires the Adobe Acrobat Reader program for viewing. Acrobat Reader freeware is available at www.adobe.com.

Note. When printing pages from the documentation PDFs, de-select Fit to Page if it is selected in your print dialog. Otherwise pages may print with slightly smaller margins.

Technical Support

An Alcatel service agreement brings your company the assurance of 7x24 no-excuses technical support. You'll also receive regular software updates to maintain and maximize your Alcatel product's features and functionality and on-site hardware replacement through our global network of highly qualified service delivery partners. Additionally, with 24-hour-a-day access to Alcatel's Service and Support web page, you'll be able to view and update any case (open or closed) that you have reported to Alcatel's technical support, open a new case or access helpful release notes, technical bulletins, and manuals. For more information on Alcatel's Service Programs, see our web page at eservice.ind.alcatel.com, call us at 1-800-995-2696, or email us at support@ind.alcatel.com.

1 Configuring Ethernet Ports

The Ethernet software is responsible for a variety of functions that support the Ethernet Network Interface (ENI) and Gigabit Network Interface (GNI) modules. These functions include diagnostics, software loading, initialization, configuration of line parameters, gathering statistics, and responding to administrative requests from SNMP or CLI.

In This Chapter

This chapter describes your switch's Ethernet port parameters and how to configure them through the Command Line Interface (CLI). CLI Commands are used in the configuration examples. For more details about the syntax of commands, see the *OmniSwitch CLI Reference Guide.*

Configuration procedures described in this chapter include:

- "Setting Trap Port Link Messages" on page 1-7
- "Setting Flow Control" on page 1-8
- "Setting Flow Control Wait Time" on page 1-9
- "Setting Interface Line Speed" on page 1-10
- "Configuring Duplex Mode" on page 1-11
- "Enabling and Disabling Interfaces" on page 1-11
- "Configuring Inter-frame Gap Values" on page 1-12
- "Resetting Statistics Counters" on page 1-13
- "Configuring Frame Size" on page 1-13
- "Configuring Flood Rates" on page 1-16
- "Configuring a Port Alias" on page 1-18
- "Configuring Auto Negotiation, Crossover, and Flow Control Settings" on page 1-18

For information about CLI commands that can be used to view Ethernet port parameters, see the *OmniSwitch CLI Reference Guide*.

Ethernet Specifications

IEEE Standards Supported	802.3 Carrier Sense Multiple Access with Collision Detection (CSMA/CD)
Ports Supported	Ethernet (10 Mbps) Fast Ethernet (100 Mbps) Gigabit Ethernet (1 Gb/1000 Mbps) 10 Gigabit Ethernet (10 Gb/10000 Mbps).
Switching/Routing Support	Layer 2 Switching/Layer 3 Routing
Backbone Support	Fast Ethernet, Gigabit Ethernet, and 10 Gigabit Ethernet ports
Port Mirroring Support	Fast Ethernet and Gigabit Ethernet ports
802.1Q Hardware Tagging	Fast Ethernet, Gigabit Ethernet, and 10 Gigabit Ethernet ports
Jumbo Frame Configuration	Supported on Gigabit Ethernet and 10 Gigabit Ethernet ports

Ethernet Port Defaults

Parameter Description	Command	Default Value/Comments
Trap Port Link Messages	**trap port link**	Disabled
Flow Control	**flow**	Disabled
Flow Control Wait Time	**flow wait time**	0 microseconds
Interface Line Speed	**interfaces speed**	Auto (all modules except for the OS7-ENI-FM12, OS7-GNI-U12, OS8-GNI-U24); 100 Mbps (OS7-ENI-FM12); 1 Gbps (OS7-GNI-U12, OS8-GNI-U24)
Duplex Mode	**interfaces duplex**	Auto (all modules except for the OS7-ENI-FM12, OS7-GNI-U12, OS8-GNI-U24, OS8-10GNI-UR1); Full (OS7-ENI-FM12, OS7-GNI-U12, OS8-GNI-U24, OS8-10GNI-UR1)
Interface Configuration	**interfaces admin**	Up (Enabled)
Inter-Frame Gap	**interfaces ifg**	12 bytes
Maximum Frame Size Configuration Status	**interfaces long**	Disable (Ethernet) Enable (Gigabit Ethernet)
Maximum Frame Size	**interfaces max frame**	1553 Ethernet packets 9216 Gigabit Ethernet packets
Minimum Frame Size	**interfaces runt**	Disable
Minimum Frame Size Configuration	**interfaces runtsize**	64 bytes

Parameter Description	Command	Default Value/Comments
Maximum Flood Rate (Broadcast Traffic)	**interfaces flood**	Enable
Maximum Flood Rate (Multicast Traffic)	**interfaces flood multicast**	Disable
Peak Flood Rate Configuration	**interfaces flood rate**	47 Mbps (Fast Ethernet) 496 Mbps (Gigabit Ethernet)
Auto negotiation	**interfaces autoneg**	Enable except for the OS7-ENI-FM12 where it is disabled
Crossover	**interfaces crossover**	Auto for all copper ports except for the GBIC-C for the OS7-GNI-U2; Disable for all fiber modules
Flow (pause)	**interfaces flow**	Enable for all modules except for the OS7-ENI-FM12

Configuring Ethernet Ports Tutorial

This tutorial describes typical steps involved in configuring an Ethernet port. This example presumes that slot 1, port 1 is an Ethernet port.

1 This step configures the line speed for slot 1, port 1 with the **interfaces speed** command. For example, to set the interface line speed for slot 1, port 1 to 100 Mbps enter:

```
-> interfaces 1/1 speed 100
```

2 This step configures the interface duplex mode for the interface in slot 1, port 1 with the **interfaces duplex** command. In full duplex mode, the interface transmits and receives data simultaneously. In half duplex mode, the interface can either transmit or receive data at a given time. For example, to set the interface duplex mode for slot 1, port 1 to full duplex enter:

```
-> interfaces 1/1 duplex full
```

Note. Duplex mode must be set to full duplex in order to set Flow Control (described below).

3 This step configures flow control wait time for this port with the **flow wait time** command. Flow control wait time specifies the amount of time (in microseconds) that the transmitting device waits before resuming transmission of data packets to the receiving device. For example, to configure the flow control wait time for slot 1, port 1 to 46 microseconds enter:

```
-> flow 1/1 wait time 46
```

4 This step enables the maximum frame size for this port with the **interfaces long** command. By setting this value to enable, the maximum frame size can be changed. For example, to enable the maximum frame size for slot 1, port 1 enter:

```
-> interfaces 1/1 long enable
```

5 This step configures the maximum frame size (in bytes) for this interface with the **interfaces max frame** command. Values for this command range from 1518 to 9216 bytes for Gigabit Ethernet ports. For example, to configure the maximum frame size for slot 1, port 1 to 1550 bytes enter:

```
-> interfaces 1/1 max frame 1550
```

6 This step enables the minimum frame size for this port with the **interfaces runt** command. For example, to enable the minimum frame size for slot 1, port 1 enter:

```
-> interfaces 1/1 runt enable
```

7 This step configures the minimum frame size (in bytes) for this interface with the **interfaces runtsize** command. Values range from 0 to 64 bytes. For example, to configure the minimum frame size for slot 1, port 1 to 64 bytes enter:

```
-> interfaces 1/1 runtsize 64
```

Note. *Optional.* To verify the Ethernet port configuration, use the **show interfaces** command. The display is similar to the one shown below, and provides additional statistics about received and transmitted bytes and frames.

```
Slot/Port  1/1 :
  Operational Status     : down,
  Type                   : Fast Ethernet,
  MAC address            : 00:d0:95:12:ed:04,
  BandWidth (Megabits)   : 100,                 Duplex           : Full,
  Long Accept            : Disable,             Runt Accept      : Disable,
  Long Frame Size(Bytes) : 1518,                Runt Size(Bytes) : 64
    Input :
      Bytes Received    :                    0,
      Lost Frames       :                    0,
      Unicast Frames    :                    0,
      Broadcast Frames  :                    0,
      Multicast Frames  :                    0,
      UnderSize Frames  :                    0,
      OverSize Frames   :                    0,
      Collision Frames  :                    0,
      Error Frames      :                    0,
      CRC Error Frames  :                    0,
      Alignments Error  :                    0
    Output :
      Bytes transmitted :                    0,
      Lost Frames       :                    0,
      Unicast Frames    :                    0,
      Broadcast Frames  :                    0,
      Multicast Frames  :                    0,
      UnderSize Frames  :                    0,
      OverSize Frames   :                    0,
      Collision Frames  :                    0,
      Error Frames      :                    0
```

For more information about available show commands, refer to the *OmniSwitch CLI Reference Guide.*

Ethernet Ports Overview

This chapter describes the Ethernet software CLI commands used for configuring and monitoring your switch's Ethernet port parameters. These commands allow you to handle administrative or port-related requests to and from SNMP, the CLI, or WebView. The Ethernet software supports the following Ethernet NI modules.

NI Module	Port Number/Type	User-Specified Port Speed (Mbps) Supported	User-Specified Duplex Supported	Auto Negotiation Supported?
OS7-ENI-C24	24 Copper twisted pair (RJ-45)	auto/10/100	auto/full/half	Yes
OS7-ENI-P24	24 copper power in-line twisted pair (RJ-45)	auto/10/100	auto/full/half	Yes
OS7-ENI-FM12	12 multimode fiber optic (MT-RJ)	100	full/half	No
OS7-GNI-C12	12 high density copper twisted pair (RJ-45)	auto/10/100/1000	auto/full/half	Yes
OS7-GNI-U2	Up to 2 wire-rate fiber LC and/or copper twisted pair (RJ-45)	1000	full	Yes
OS7-GNI-U12	Up to 12 high-density fiber LC ports	1000	full	Yes
OS8-ENI-C24	24 Copper twisted pair (RJ-45)	auto/10/100	auto/full/half	Yes
OS8-GNI-C8	8 wire-rate 8 copper twisted pair (RJ-45)	1000	full	Yes
OS8-GNI-C24	24 high-density copper twisted pair (RJ-45)	auto/10/100/1000	auto/full/half	Yes
OS8-GNI-U8	Up to 8 wire-rate fiber LC ports	1000	full	Yes
OS8-GNI-U24	Up to 24 high-density LC ports	1000	full	Yes
OS8-10GNI-UR1	1 (redundant) wire-rate SC port	10000	full	Yes

Fast Ethernet, Gigabit Ethernet, and 10 Gigabit Ethernet switching modules can be used as backbone links, with Gigabit Ethernet an 10 Gigabit Ethernet modules offering additional support for high-speed servers. All modules support 802.1Q hardware tagging for enhanced compatibility. And all Gigabit and 10 Gigabit modules support jumbo frame configuration.

Setting Ethernet Port Parameters

The following procedures are used to configure Ethernet port parameters on your switch. Please refer to the table on page 1-6 for Network Interface (NI) port types.

Setting Trap Port Link Messages

The **trap port link** command can be used to enable or disable (the default) trap port link messages on a specific port, a range of ports, or all ports on an NI module. When enabled, a trap message will be displayed on a Network Management Station (NMS) whenever the port state has changed.

Enabling Trap Port Link Messages

To enable trap port link messages on an entire NI, enter **trap** followed by the slot number and **port link enable**. For example, to enable trap port link messages on all ports on slot 2 enter:

```
-> trap 2 port link enable
```

To enable trap port link messages on a single port enter **trap** followed by the slot number, a slash (/), the port number, and **port link enable**. For example, to enable trap port link messages on slot 2 port 3 enter:

```
-> trap 2/3 port link enable
```

To enable trap port link messages on a range of ports enter **trap** followed by the slot number, a slash (/), the first port number, a hyphen (-), the last port number, and **port link enable**. For example, to enable trap port link messages ports 3 through 5 on slot 2 enter:

```
-> trap 2/3-5 port link enable
```

Disabling Trap Port Link Messages

To disable trap port link messages on an entire NI, enter **trap** followed by the slot number and **port link disable**. For example, to disable trap port link messages on all ports on slot 2 enter:

```
-> trap 2 port link disable
```

To disable trap port link messages on a single port enter **trap** followed by the slot number, a slash (/), the port number, and **port link disable**. For example, to disable trap port link messages on slot 2 port 3 enter:

```
-> trap 2/3 port link disable
```

To disable trap port link messages on a range of ports enter **trap** followed by the slot number, a slash (/), the first port number, a hyphen (-), the last port number, and **port link disable**. For example, to disable trap port link messages ports 3 through 5 on slot 2 enter:

```
-> trap 2/3-5 port link disable
```

Setting Flow Control

The **flow** command can be used to enable or disable (the default) flow control on a specific port, a range of ports, or all ports on an entire NI module. When the buffers on a receiving device are full, flow control transmits pause frames to the remote link partner to delay transmission. Also, the local port can delay the transmission of data if the remote link partner transmits pause frames. If auto-negotiation is implemented and enabled for the interface, the pause mode for this interface is determined by auto-negotiation.

Note. Although this command is still supported Alcatel recommends the using the **interfaces flow** command, which is described in "Enabling and Disabling Flow" on page 1-21, instead.

Enabling Flow Control

To enable flow control on an entire NI, enter **flow** followed by the slot number. For example, to enable flow control on slot 2 enter:

```
-> flow 2
```

To enable flow control on a single port, enter **flow** followed by the slot number, a slash (/), and the port number. For example, to enable flow control on port 3 on slot 2 enter:

```
-> flow 2/3
```

To enable flow control on a range of ports, enter **flow** followed by the slot number, a slash (/), the first port number, a hyphen, and the last port number. For example, to enable flow control on ports 1 through 3 on slot 2 enter:

```
-> flow 2/1-3
```

As an option, you can document the interface type by entering **ethernet**, **fastethernet**, or **gigaethernet** before the slot number. For example to enable flow control on the interface on slot 2 port 3 and documents the interface type as Fast Ethernet enter:

```
-> flow fastethernet 2/3
```

Disabling Flow Control

To disable flow control on an entire NI, enter **no flow** followed by the slot number. For example, to disable flow control on slot 2 enter:

```
-> no flow 2
```

To disable flow control on a single port, enter **no flow** followed by the slot number, a slash (/), and the port number. For example, to disable flow control on port 3 on slot 2 enter:

```
-> no flow 2/3
```

To disable flow control on a range of ports, enter **no flow** followed by the slot number, a slash (/), the first port number, a hyphen, and the last port number. For example, to disable flow control on ports 1 through 3 on slot 2 enter:

```
-> no flow 2/1-3
```

As an option, you can document the interface type by entering **ethernet**, **fastethernet**, or **gigaethernet** before the slot number. For example to disable flow control on the interface on slot 2 port 3 and documents the interface type as Fast Ethernet enter:

```
-> no flow fastethernet 2/3
```

Setting Flow Control Wait Time

By default, the flow control wait time is 0 microseconds. Use the **flow wait time** command to configure flow control wait time on a specific port, a range of ports, or all ports on an NI module. When configured, flow control wait time specifies the amount of time (in microseconds) that the transmitting device waits before resuming transmission of data packets to the receiving device. The valid range is 0 to 30000 microseconds. (The flow control wait time on 10 Mbps ports is not configurable.)

Note. If auto-negotiation is implemented and enabled for the interface, the Pause mode for this interface is determined by Auto-negotiation and Full-duplex.

Configuring the Flow Control Wait Time

To configure flow control wait time for an entire NI, enter **flow** followed by the slot number, **wait**, and the desired wait time in microseconds. For example, to configure a flow control wait time of 96 microseconds on slot 2 enter:

```
-> flow 2 wait 96
```

Note. Setting the flow control wait time to zero (0) accomplishes the same function as the **flow no wait** command (restoring flow control wait time). See "Restoring the Flow Control Wait Time" on page 1-10 for more information.

To configure flow control wait time for a single port, enter **flow** followed by the slot number, a slash (/), the port number, **wait**, and the desired wait time in microseconds. For example, to configure a flow control wait time of 96 microseconds on slot 2 port 3 enter:

```
-> flow 2/3 wait 96
```

To configure flow control wait time for a range of ports, enter **flow** followed by the slot number, a slash (/), the first port number, a hyphen (**-**), the last port number, **wait**, and the desired wait time in microseconds. For example, to configure a flow control wait time of 96 microseconds on ports 1 through 3 on slot 2 enter:

```
-> flow 2/1-3 wait 96
```

As an option, you can document the interface type by entering **ethernet**, **fastethernet**, or **gigaethernet** before the slot number. For example, to configure the flow control wait time as 96 microseconds on slot 2 port 3 and document the interface type as Fast Ethernet enter:

```
-> flow fastethernet 2/3 wait 96
```

Restoring the Flow Control Wait Time

To restore the flow control wait time (i.e., set it back to 0) for an entire NI, enter **flow** followed by the slot number and **no wait**. For example, to restore the flow control wait time to 0 seconds on slot 2 enter:

```
-> flow 2 no wait
```

To restore the flow control wait time (i.e., set it back to 0) for a single port, enter **flow** followed by the slot number, a slash (/), the port number, and **no wait**. For example, to restore the flow control wait time of 0 seconds on slot 2 port 3 enter:

```
-> flow 2/3 no wait
```

To restore the flow control wait time (i.e., set it back to 0) for a range of ports, enter **flow** followed by the slot number, a slash (/), the first port number, a hyphen (**-**), the last port number, and **no wait**. For example, to restore the flow control wait time of 0 seconds on ports 1 through 3 on slot 2 enter:

```
-> flow 2/1-3 no wait
```

As an option, you can document the interface type by entering **ethernet**, **fastethernet**, or **gigaethernet** before the slot number. For example, to restore the flow control wait time of 0 seconds on slot 2 port 3, and document the interface type as Fast Ethernet enter:

```
-> flow fastethernet 2/3 no wait
```

Setting Interface Line Speed

The **interfaces speed** command is used to set the line speed on a specific port, a range of ports, or all ports on an entire NI module to **10** (10 Mbps Ethernet), **100** (100 Mbps Fast Ethernet), **1000** (1000 Mbps Gigabit Ethernet), **10000** (10 Gigabit Ethernet), **auto** (auto-sensing), a maximum of 100 Mbps (**max 100**), or a maximum of 1000 Mbps (**max 1000**). The auto setting automatically detects and matches the line speed of the attached device.

Note. Available settings for this command depend on the available line speeds of your hardware interface. See the table on page 1-6 for more information.

To set the line speed on an entire NI enter **interfaces** followed by the slot number and the desired speed. For example, to set slot 2 to 100 Mbps enter:

```
-> interfaces 2 speed 100
```

To set the line speed on a single port enter **interfaces** followed by the slot number, a slash (/), the port number, and the desired speed. For example, to set the line speed on slot 2 port 3 at 100 Mbps enter:

```
-> interfaces 2/3 speed 100
```

To set the line speed on a range of ports enter **interfaces** followed by the slot number, a slash (/), the first port number, a hyphen (**-**), the last port number, and the desired speed. For example, to set the line speed on ports 1 through 3 on slot 2 at 100 Mbps enter:

```
-> interfaces 2/1-3 speed 100
```

As an option, you can document the interface type by entering **ethernet**, **fastethernet**, or **gigaethernet** before the slot number. For example, to configure the line speed on slot 2 port 3 at 100 Mbps, and document the interface type as Fast Ethernet enter:

```
-> interfaces fastethernet 2/3 speed 100
```

Configuring Duplex Mode

The **interfaces duplex** command is used to configure the duplex mode on a specific port, a range of ports, or all ports on an NI module to **full** (full duplex mode), **half** (half duplex mode), **auto** (auto-negotiation). In full duplex mode, the interface transmits and receives data simultaneously. In half duplex mode, the interface can only transmit or receive data at a given time. (Available settings for this command depend on the available duplex modes of your hardware interface. See the table on page 1-6 for more information.)

Note. The **Auto** option causes the switch to advertise all available duplex modes (half/full/both) for the port during autonegotiation.

To configure the duplex mode on an entire slot enter **interfaces** followed by the slot number, **duplex**, and the desired duplex setting (**auto**, **full**, or **half**). For example, to set the duplex mode on slot 2 to full enter:

```
-> interfaces 2 duplex full
```

To configure the duplex mode on a single port enter **interfaces** followed by the slot number, a slash (/), the port number, **duplex**, and the desired duplex setting (**auto**, **full**, or **half**). For example, to set the duplex mode on port 3 on slot 2 to full enter:

```
-> interfaces 2/3 duplex full
```

To configure the duplex mode on a range of ports enter **interfaces** followed by the slot number, a slash (/), the first port number, a hyphen (**-**), the last port number, **duplex**, and the desired duplex setting (**auto**, **full**, or **half**). For example, to set the duplex mode on ports 1 through 3 on slot 2 to full enter:

```
-> interfaces 2/1-3 duplex full
```

As an option, you can document the interface type by entering **ethernet**, **fastethernet**, or **gigaethernet** before the slot number. For example, to set the duplex mode on port 3 on slot 2 and document the port as Fast Ethernet enter:

```
-> interfaces fastethernet 2/3 duplex full
```

Enabling and Disabling Interfaces

The **interfaces admin** command is used to enable (the default) or disable a specific port, a range of ports, or all ports on an NI module.

To enable or disable an entire slot enter **interfaces** followed by the slot number, **admin**, and the desired administrative setting (either **up** or **down**). For example, to administratively disable slot 2 enter:

```
-> interfaces 2 admin down
```

To enable or disable a single port enter **interfaces** followed by the slot number, a slash (/), the port number, **admin**, and the desired administrative setting (either **up** or **down**). For example, to administratively disable port 3 on slot 2 enter:

```
-> interfaces 2/3 admin down
```

To enable or disable a range of ports enter **interfaces** followed by the slot number, a slash (/), the first port number, a hyphen (**-**), the last port number, **admin**, and the desired administrative setting (either **up** or **down**). For example, to administratively disable ports 1 through 3 on slot 2 enter:

```
-> interfaces 2/1-3 admin down
```

As an option, you can document the interface type by entering **ethernet**, **fastethernet**, or **gigaethernet** before the slot number. For example, to administratively disable port 3 on slot 2 and document the port as Fast Ethernet:

```
-> interfaces fastethernet 2/3 admin down
```

Configuring Inter-frame Gap Values

Inter-frame gap is a measure of the minimum idle time between the end of one frame transmission and the beginning of another. By default, the inter-frame gap is 12 bytes. The **interfaces ifg** command can be used to configure the inter-frame gap value (in bytes) on a specific port, a range of ports, or all ports on an NI module. Values for this command range from 9 to 12 bytes.

Note. This command is only valid on Gigabit ports.

To configure the inter-frame gap on an entire slot enter **interfaces**, followed by the slot number, **ifg**, and the desired inter-frame gap value. For example, to set the inter-frame gap value on slot 2 to 10 bytes enter:

```
-> interfaces 2 ifg 10
```

To configure the inter-frame gap on a single port enter **interfaces**, followed by the slot number, a slash (/), the port number, **ifg**, and the desired inter-frame gap value. For example, to set the inter-frame gap value on port 3 on slot 2 to 10 bytes enter:

```
-> interfaces 2/3 ifg 10
```

To configure the inter-frame gap on a range of ports enter **interfaces**, followed by the slot number, a slash (/), the first port number, a hyphen (-), the last port number, **ifg**, and the desired inter-frame gap value. For example, to set the inter-frame gap value on ports 1 through 3 on slot 2 to 10 bytes enter:

```
-> interfaces 2/1-3 ifg 10
```

As an option, you can document the interface type by entering **ethernet**, **fastethernet**, or **gigaethernet** before the slot number. For example, to set the inter-frame gap value on port 3 on slot 2 to 10 bytes and document the port as Gigabit Ethernet enter:

```
-> interfaces gigaethernet 2/3 ifg 10
```

Note. Since the **interfaces ifg** command is only supported on Gigabit interfaces only the **gigaethernet** keyword should be used.

Resetting Statistics Counters

The **interfaces no l2 statistics** command is used to reset all Layer 2 statistics counters on a specific port, a range of ports, or all ports on an NI module.

To reset Layer 2 statistics on an entire slot enter **interfaces** followed by the slot number and **no l2 statistics**. For example, to reset all Layer 2 statistics counters on slot 2 enter:

```
-> interfaces 2 no l2 statistics
```

To reset Layer 2 statistics on a single port enter **interfaces** followed by the slot number, a slash (/), the port number, and **no l2 statistics**. For example, to reset all Layer 2 statistics counters on port 3 on slot 2 enter:

```
-> interfaces 2/3 no l2 statistics
```

To reset Layer 2 statistics on a range of ports enter **interfaces** followed by the slot number, a slash (/), the first port number, a hyphen (**-**), the last port number, and **no l2 statistics**. For example, to reset all Layer 2 statistics counters on ports 1 through 3 on slot 2 enter:

```
-> interfaces 2/1-3 no l2 statistics
```

As an option, you can document the interface type by entering **ethernet**, **fastethernet**, or **gigaethernet** before the slot number. For example, to reset all Layer 2 statistics counters on port 3 on slot 2 and document the port as Fast Ethernet:

```
-> interfaces fastethernet 2/3 no l2 statistics
```

Note. The **show interfaces**, **show interfaces accounting**, and **show interfaces counters** commands can be used to display Layer 2 statistics (e.g., input and output errors, deferred frames received, unicast packets transmitted). For information on using these commands, see Chapter 1, "Ethernet Port Commands," in the *OmniSwitch CLI Reference Guide.*

Configuring Frame Size

The following subsections describe how to enable and disable maximum frame size configuration (see "Enabling and Disabling Maximum Frame Size Configuration" on page 1-13), configuring maximum frame size (see "Configuring Maximum Frame Sizes" on page 1-14), enabling and disabling minimum frame size configuration (see "Enabling and Disabling Minimum Frame Size Configuration" on page 1-15), and configuring minimum fame size (see "Configuring Minimum Frame Sizes" on page 1-15).

Enabling and Disabling Maximum Frame Size Configuration

The **interfaces long** command can be used to enable (the default) or disable maximum frame size configuration on a specific port, a range of ports, or all ports on an NI module. By specifying the setting for this command as enable, the maximum frame size can be changed.

To enable or disable maximum frame size configuration on an entire slot enter **interfaces** followed by the slot number, **long**, and either **enable** or **disable**. For example, to enable maximum frame size configuration on slot 2 enter:

```
-> interfaces 2 long enable
```

To enable or disable maximum frame size configuration on a single port enter **interfaces** followed by the slot number, a slash (/), the port number, **long**, and either **enable** or **disable**. For example, to enable maximum frame size configuration on port 3 on slot 2 enter:

```
-> interfaces 2/3 long enable
```

To enable or disable maximum frame size configuration on a range of ports enter **interfaces** followed by the slot number, a slash (/), the first port number, a hyphen (**-**), the last port number, **long**, and either **enable** or **disable**. For example, to enable maximum frame size configuration on ports 1 through 3 on slot 2 enter:

```
-> interfaces 2/1-3 long enable
```

As an option, you can document the interface type by entering **ethernet**, **fastethernet**, or **gigaethernet** before the slot number. For example, to enable maximum frame size configuration on port 3 on slot 2 and document the port as Gigabit Ethernet enter:

```
-> interfaces gigaethernet 2/3 long enable
```

Note. The maximum frame size value is changed internally if the **interfaces long disable** command is used.

Configuring Maximum Frame Sizes

The **interfaces max frame** command can be used to configure the maximum frame size (in bytes) on a specific port, a range of ports, or all ports on an NI module. Maximum values for this command range from 1518 to 9216 bytes (Gigabit Ethernet packets) for Gigabit Ethernet ports.

To configure the maximum frame size on an entire slot enter **interfaces** followed by the slot number, **max frame**, and the frame size in bytes. For example, to set the maximum frame size on slot 2 to 9216 bytes enter:

```
-> interfaces 2 max frame 9216
```

To configure the maximum frame size on a single port enter **interfaces** followed by the slot number, a slash (/), the port number, **max frame**, and the frame size in bytes. For example, to set the maximum frame size on port 3 on slot 2 to 9216 bytes enter:

```
-> interfaces 2/3 max frame 9216
```

To configure the maximum frame size on a range of ports enter **interfaces** followed by the slot number, a slash (/), the first port number, a hyphen (**-**), the last port number, **max frame**, and the frame size in bytes. For example, to set the maximum frame size on ports 1 through 3 on slot 2 to 9216 bytes enter:

```
-> interfaces 2/1-3 max frame 9216
```

As an option, you can document the interface type by entering **ethernet**, **fastethernet**, or **gigaethernet** before the slot number. For example, to set the maximum frame size on port 3 on slot 2 to 9216 bytes and document the port as Gigabit Ethernet enter:

```
-> interfaces gigaethernet 2/3 max frame 9216
```

Enabling and Disabling Minimum Frame Size Configuration

The **interfaces runt** command can be used to enable or disable (the default) minimum frame size (in bytes) configuration on a specific port, a range of ports, or all ports on an NI module. By specifying the setting for this command as enable, the minimum frame size can be changed.

To enable or disable minimum frame size configuration on an entire slot enter **interfaces** followed by the slot number, **runt**, and either **enable** or **disable**. For example, to enable minimum size configuration on slot 2 enter:

```
-> interfaces 2 runt enable
```

To enable or disable minimum frame size configuration on a single port enter **interfaces** followed by the slot number, a slash (/), the port number, **runt**, and either **enable** or **disable**. For example, to enable minimum size configuration on port 3 on slot 2 enter:

```
-> interfaces 2/3 runt enable
```

To enable or disable minimum frame size configuration on a range of ports enter **interfaces** followed by the slot number, a slash (/), the first port number, a hyphen (**-**), the last port number, **runt**, and either **enable** or **disable**. For example, to enable minimum size configuration on ports 1 through 3 on slot 2 enter:

```
-> interfaces 2/1-3 runt enable
```

As an option, you can document the interface type by entering **ethernet**, **fastethernet**, or **gigaethernet** before the slot number. For example, to enable minimum size configuration on port 3 on slot 2 and document the port as Gigabit Ethernet enter:

```
-> interfaces gigaethernet 2/3 runt enable
```

Note. The minimum frame size value is automatically changed internally if the **interfaces runt disable** command is used.

Configuring Minimum Frame Sizes

The **interfaces runtsize** command can be used to configure the minimum frame size on a specific port, a range of ports, or all ports on an NI module from 0 to 64 bytes.

To configure the minimum frame size on an entire slot enter **interfaces** followed by the slot number, **runtsize**, and the frame size in bytes. For example, to configure the minimum frame size on slot 2 as 32 bytes enter:

```
-> interfaces 2 runtsize 32
```

To configure the minimum frame size on a single port enter **interfaces** followed by the slot number, a slash (/), the port number, **runtsize**, and the frame size in bytes. For example, to configure the minimum frame size on port 3 on slot 2 as 32 bytes enter:

```
-> interfaces 2/3 runtsize 32
```

To configure the minimum frame size on a range of ports enter **interfaces** followed by the slot number, a slash (/), the first port number, a hyphen (**-**), the last port number, **runtsize**, and the frame size in bytes. For example, to configure the minimum frame size on ports 1 through 3 on slot 2 as 32 bytes enter:

```
-> interfaces 2/1-3 runtsize 32
```

As an option, you can document the interface type by entering **ethernet**, **fastethernet**, or **gigaethernet** before the slot number. For example, to configure the minimum frame size on port 3 on slot 2 as 32 bytes and document the port as Gigabit Ethernet enter:

```
-> interfaces gigaethernet 2/3 runtsize 32
```

Configuring Flood Rates

The following subsections describe how to enable the maximum flood rate (see "Enabling the Maximum Flood Rate" on page 1-16), enable the maximum flood rate for multicast traffic (see "Enabling Maximum Flood Rate for Multicast Traffic" on page 1-16), and how to configure the flood rate on an entire NI module, a specific port, or a range of ports (see "Configuring Flood Rate Values" on page 1-17).

Enabling the Maximum Flood Rate

The **interfaces flood** command can be used to enable the maximum flood rate for an NI module. Note that only one slot can be configured at a time. You cannot configure specific ports or ranges of ports.

Note. To enable flood multicasting on an interface, see "Enabling Maximum Flood Rate for Multicast Traffic" on page 1-16.

To enable the maximum flood rate on a slot enter **interfaces** followed by the slot number and **flood**. For example, to enable the maximum flood rate on slot 2 enter:

```
-> interfaces 2 flood
```

As an option, you can document the interface type by entering **ethernet**, **fastethernet**, or **gigaethernet** before the slot number. For example, to enable the maximum flood rate on slot 2 enter and document the slot as Gigabit Ethernet enter:

```
-> interfaces gigaethernet 2 flood
```

Enabling Maximum Flood Rate for Multicast Traffic

The **interfaces flood multicast** command can be used to enable the maximum flood rate for multicast traffic for an NI module. Note that only one slot can be configured per command. You cannot configure specific ports or ranges of ports.

Note. To enable maximum flood rate on an interface and to disable any flood multicast configuration use the **interface flood** command, which is described on "Enabling the Maximum Flood Rate" on page 1-16.

To enable the maximum flood rate for multicast traffic on a slot enter **interfaces** followed by the slot number and **flood multicast**. For example, to enable the maximum flood rate for multicast traffic on slot 2 enter:

```
-> interfaces 2 flood multicast
```

As an option, you can document the interface type by entering **ethernet**, **fastethernet**, or **gigaethernet** before the slot number. For example, to enable the maximum flood rate for multicast traffic on slot 2 enter and document the slot as Gigabit Ethernet enter:

```
-> interfaces gigaethernet 2 flood multicast
```

Note. Enabling the maximum multicast flood rate with the **interfaces flood multicast** command will limit IP Multicast Switching (IPMS) and non-IPMS multicast traffic.

Configuring Flood Rate Values

The **interfaces flood rate** command can be used to configure the peak flood rate value on a specific port, a range of ports, or all ports on an NI module in megabits per second, ranging from 0 to 8 Mbps for Ethernet, 0 to 98 Mbps for Fast Ethernet, or 0 to 996 Mbps for Gigabit Ethernet.

Note. The flood rate cannot be higher than line speed.

To configure the peak flood rate for an entire slot enter **interfaces** followed by the slot number, **flood rate**, and the flood rate in bytes. For example, to configure the peak flood rate on slot 2 as 47 bytes enter:

```
-> interfaces 2 flood rate 47
```

To configure the peak flood rate for a single port enter **interfaces** followed by the slot number, a slash (**/**), the port number, **flood rate**, and the flood rate in bytes. For example, to configure the peak flood rate on port 3 on slot 2 as 47 bytes enter:

```
-> interfaces 2/3 flood rate 47
```

To configure the peak flood rate for a range of ports enter **interfaces** followed by the slot number, a slash (**/**), the first port number, a hyphen (**-**), the last port number, **flood rate**, and the flood rate in bytes. For example, to configure the peak flood rate on ports 1 through 3 on slot 2 as 47 bytes enter:

```
-> interfaces 2/1-3 flood rate 47
```

As an option, you can document the interface type by entering **ethernet**, **fastethernet**, or **gigaethernet** before the slot number. For example, to configure the peak flood rate on port 3 on slot 2 as 47 bytes and document the port as Gigabit Ethernet enter:

```
-> interfaces gigaethernet 2/3 flood rate 47
```

Configuring a Port Alias

The **interfaces alias** command is used to configure an alias (i.e., description) for a single port. (You cannot configure an entire NI module or a range of ports.) To use this command enter **interfaces** followed by the slot number, a slash (/), the port number, **alias**, and the text description, which can be up to 40 characters long.

For example, to configure an alias of "ip_phone1" for port 3 on slot 2 enter:

```
-> interfaces 2/3 alias ip_phone1
```

Note. Spaces must be contained within quotes (e.g., "IP Phone 1").

As an option, you can document the interface type by entering **ethernet**, **fastethernet**, or **gigaethernet** before the slot number. For example, to configure an alias of "ip_phone1" for port 3 on slot 2 and document the port as Fast Ethernet enter:

```
-> interfaces fastethernet 2/3 alias ip_phone1
```

Configuring Auto Negotiation, Crossover, and Flow Control Settings

The following subsections describe how to enable and disable auto negotiation (see "Enabling and Disabling Auto Negotiation" on page 1-18), configuring crossover settings (see "Configuring Crossover Settings" on page 1-20), and configuring flow control (see "Enabling and Disabling Flow" on page 1-21). Default values and valid configurations, which vary depending on the Network Interface (NI) module type, are also described in the following subsections.

Enabling and Disabling Auto Negotiation

By default, auto negotiation is enabled on all NI modules except for the OS7-ENI-FM12. (It is disabled by default and not configurable on this module.) To enable or disable auto negotiation on a single port, a range of ports, or an entire NI use the **interfaces autoneg** command. If auto negotiation is auto MDIX, flow control, auto speed, and auto duplex are not accepted. (See "Configuring Crossover Settings" on page 1-20 and "Enabling and Disabling Flow" on page 1-21 for more information).

To enable or disable auto negotiation on an entire slot enter **interfaces** followed by the slot number, **autoneg**, and either **enable** or **disable**. For example, to enable auto negotiation on slot 2 enter:

```
-> interfaces 2 autoneg enable
```

To enable or disable auto negotiation on a single port enter **interfaces** followed by the slot number, a slash (/), the port number, **autoneg**, and either **enable** or **disable**. For example, to enable auto negotiation on port 3 on slot 2 enter:

```
-> interfaces 2/3 autoneg enable
```

To enable or disable auto negotiation on a range of ports enter **interfaces** followed by the slot number, a slash (/), the first port number, a hyphen (-), the last port number, **autoneg**, and either **enable** or **disable**. For example, to enable auto negotiation on ports 1 through 3 on slot 2 enter:

```
-> interfaces 2/1-3 autoneg enable
```

As an option, you can document the interface type by entering **ethernet**, **fastethernet**, or **gigaethernet** before the slot number. For example, to enable auto negotiation on port 3 on slot 2 and document the port as Gigabit Ethernet enter:

```
-> interfaces gigaethernet 2/3 autoneg enable
```

Please note a link will not be established on any copper Ethernet port if any one of the following is true:

- The local port advertises 100 Mbps full duplex and the remote link partner is forced to 100 Mbps full duplex.
- The local port advertises 100 Mbps full duplex and the remote link partner is forced to 100 Mbps half duplex.
- The local port advertises 10 Mbps full duplex and the remote link partner is forced to 10 Mbps full duplex.
- The local port advertises 10 Mbps full duplex and the remote link partner is forced to 10 half duplex.

This is due to the fact that when the local device is set to auto negotiating 10/100 full duplex it senses the remote device is not auto negotiating. Therefore it resolves to Parallel Detect with Highest Common Denominator (HCD), which is “10/100 Half” according to IEEE 802.3 Clause 28.2.3.1.

However, since the local device is set to auto negotiating at 10/100 full duplex it cannot form a 10/100 Mbps half duplex link in any of the above mentioned cases. One solution is to configure the local device to auto negotiation, 10/100 Mbps, with auto or half duplex.

Auto Negotiation on OS7-GNI-C12 and OS8-GNI-C24 Modules

As shown in the table below, the valid configuration and default settings change for the high density copper Gigabit OS7-GNI-C12 and OS8-GNI-C24 modules depending if auto negotiation is enabled or disabled.

OS7-GNI-C12/OS8-GNI-C24 Auto Negotiation Combinations

Auto Negotiation Status	Valid Speed Settings (Mbps)	Valid Duplex Setting	Valid Cross-Over Setting	Valid Flow Settings
enable	auto (default), 10, 100, 1000	auto (default), full, half	auto (default), mdi, mdix	enable (default), disable
disable	10, 100 (default)	full, half (default)	mdi, mdix (default will be MDIX if the link was down or “as is” if the link was up)	enable (default), disable

Note. If you re-enable auto negotiation on the high-density copper Gigabit OS7-GNI-C12 and OS8-GNI-C24 modules it will restore the default speed, duplex, and flow settings.

Also, when auto negotiation is enabled, auto crossover is also enabled. When auto negotiation is disabled (the port is now 10/100 mode) auto crossover is disabled and the mode of cable (MDI/MDIX) is left “as is” according to what it is connected to if the link is up or set to MDIX (default) if there is no link. Note that when auto negotiation is enabled on the OS7-GNI-C12 and OS8-GNI-C24 all options (Speed: Auto, Duplex: Auto, Crossover: Auto) are advertised, *not* just what is configured.

Configuring Crossover Settings

To configure crossover settings on a single port, a range of ports, or an entire Network Interface (NI) use the **interfaces crossover** command. Defaults and valid settings for this command are shown in the table below. If auto negotiation is disabled, flow control, auto speed, and auto duplex are not accepted.

NI Module	Default Crossover Setting	Valid Crossover Settings
OS7-ENI-C24	auto	auto, mdix, mdi, disable
OS7-ENI-P24	auto	auto, mdix, mdi, disable
OS7-ENI-FM12	disable	disable
OS7-GNI-C12	auto	auto, mdix, mdi, disable
OS7-GNI-U2	disable	disable
OS7-GNI-U12	disable	disable
OS8-ENI-C24	auto	auto, mdix, mdi, disable
OS8-GNI-C8	auto	auto, mdix, mdi, disable
OS8-GNI-C24	auto	auto, mdix, mdi, disable
OS8-GNI-U8	disable	disable
OS8-GNI-U24	disable	disable
OS8-10GNI-UR1	disable	disable

Setting the crossover configuration to **auto** will configure the interface or interfaces to automatically detect crossover settings. Setting crossover configuration to **mdix** will configure the interface or interfaces for MDIX (Media Dependent Interface with Crossover), which is the standard for hubs and switches. Setting crossover to **mdi** will configure the interface or interfaces for MDI (Media Dependent Interface), which is the standard for end stations. And setting the crossover configuration to **disable** will disable crossover configuration on an interface or interfaces.

To configure crossover settings on an entire slot enter **interfaces** followed by the slot number, **crossover**, and the desired setting. (See the table on page 1-20 for valid settings.) For example, to set the crossover configuration to auto on slot 2 enter:

```
-> interfaces 2 crossover auto
```

To configure crossover settings on a single port enter **interfaces** followed by the slot number, a slash (**/**), the port number, **crossover**, and the desired setting. (See the table on page 1-20 for valid settings.) For example, to set the crossover configuration to auto on port 3 on slot 2 enter:

```
-> interfaces 2/3 crossover auto
```

To configure crossover settings on a range of ports enter **interfaces** followed by the slot number, a slash (**/**), the first port number, a hyphen (**-**), the last port number, **crossover**, and the desired setting. (See the table on page 1-20 for valid settings.) For example, to set the crossover configuration to auto on ports 1 through 3 on slot 2 enter:

```
-> interfaces 2/1-3 crossover auto
```

As an option, you can document the interface type by entering **ethernet**, **fastethernet**, or **gigaethernet** before the slot number. For example, to set the crossover configuration to auto on port 3 on slot 2 and document the port as Fast Ethernet enter:

```
-> interfaces fastethernet 2/3 crossover auto
```

Enabling and Disabling Flow

By default, flow (pause) is enabled. To enable or disable flow control on a single port, a range of ports, or an entire NI use the **interfaces flow** command. Please note that if auto negotiation is disabled then flow control will also be disabled.

To enable or disable flow control on an entire slot enter **interfaces** followed by the slot number, **flow**, and either **enable** or **disable**. For example, to enable flow control on slot 2 enter:

```
-> interfaces 2 flow enable
```

To enable or disable flow control on a single port enter **interfaces** followed by the slot number, a slash (/), the port number, **flow**, and either **enable** or **disable**. For example, to enable flow control on port 3 on slot 2 enter:

```
-> interfaces 2/3 flow enable
```

To enable or disable flow control on a range of ports enter **interfaces** followed by the slot number, a slash (/), the first port number, a hyphen (**-**), the last port number, **flow**, and either **enable** or **disable**. For example, to enable flow control on ports 1 through 3 on slot 2 enter:

```
-> interfaces 2/1-3 flow enable
```

As an option, you can document the interface type by entering **ethernet**, **fastethernet**, or **gigaethernet** before the slot number. For example, to enable flow control on port 3 on slot 2 and document the port as Fast Ethernet enter:

```
-> interfaces fastethernet 2/3 flow enable
```

Note. If auto negotiation is disabled and then later enabled on an interface, the original flow setting will then be restored.

Verifying Ethernet Port Configuration

To display information about Ethernet port configuration settings, use the **show** commands listed in the following table.

show interfaces flow control	Displays interface flow control wait time and cross-over settings.
show interfaces	Displays general interface information, such as hardware, MAC address, input and output errors.
show interfaces accounting	Displays interface accounting information.
show interfaces counters	Displays interface counters information.
show interfaces counters errors	Displays interface error frame information for Ethernet and Fast Ethernet ports.
show interfaces collisions	Displays collision statistics information for Ethernet and Fast Ethernet ports.
show interfaces status	Displays line status information.
show interfaces port	Displays port status information.
show interfaces ifg	Displays inter-frame gap values.
show interfaces flood rate	Displays peak flood rate settings.
show interfaces traffic	Displays interface traffic statistics.
show interfaces capability	Displays auto negotiation, flow, speed, duplex, and cross-over settings.
show 10gig	Displays the status of 10 Gigabit modules in an OmniSwitch 8800 chassis.

These commands can be quite useful in troubleshooting and resolving potential configuration issues or problems on your switch. For more information about the resulting displays from these commands, see the *OmniSwitch CLI Reference Guide*.

2 Managing Source Learning

Transparent bridging relies on a process referred to as *source learning* to handle traffic flow. Network devices communicate by sending and receiving data packets that each contain a source MAC address and a destination MAC address. When packets are received on switch network interface (NI) module ports, source learning examines each packet and compares the source MAC address to entries in a MAC address database table. If the table does not contain an entry for the source address, then a new record is created associating the address with the port it was learned on. If an entry for the source address already exists in the table, a new one is not created.

Packets are also filtered to determine if the source and destination address are on the same LAN segment. If the destination address is not found in the MAC address table, then the packet is forwarded to all other switches that are connected to the same LAN. If the MAC address table does contain a matching entry for the destination address, then there is no need to forward the packet to the rest of the network.

In This Chapter

This chapter describes how to manage source learning entries in the switch MAC address table (often referred to as the *forwarding* or *filtering database*) through the Command Line Interface (CLI). CLI commands are used in the configuration examples; for more details about the syntax of commands, see the *OmniSwitch CLI Reference Guide*.

Configuration procedures described in this chapter include:

- Creating a static MAC address table entry on page 2-4.
- Configuring the MAC address table aging time on page 2-6.
- Displaying MAC address table information on page 2-7.

Source Learning Specifications

RFCs supported	2674 - *Definitions of Managed Objects for Bridges with Traffic Classes, Multicast Filtering and Virtual LAN Extensions*
IEEE Standards supported	802.1Q - *Virtual Bridged Local Area Networks* 802.1D - *Media Access Control Bridges*
Number of learned MAC addresses per network interface (NI) module	32K
Number of learned MAC addresses per switch	64K

Source Learning Defaults

Parameter Description	Command	Default
Static MAC address management status	**mac-address-table**	permanent
Static MAC address operating mode	**mac-address-table**	bridging
MAC address aging timer	**mac-address-table aging-time**	300 seconds per VLAN

Sample MAC Address Table Configuration

The following steps provide a quick tutorial that will create a static MAC address and change the MAC address aging timer for VLAN 200:

Note. *Optional.* Creating a static MAC address involves specifying an address that is not already used in another static entry or already dynamically learned by the switch. To determine if the address is already known to the MAC address table, enter **show mac-address-table**. If the address does not appear in the **show mac-address-table** output, then it is available to use for configuring a static MAC address entry. For example,

```
-> show mac-address-table
Legend: Mac Address: * = address not valid

   Vlan      Mac Address          Type       Protocol    Operation    Interface
  ------+-------------------+--------------+-----------+------------+-----------
     1   00:00:00:00:00:01     learned        0800        bridging      8/ 1
     1   00:d0:95:6a:73:9a     learned        aaaa0003    bridging     10/23
Total number of Valid MAC addresses above = 2
```

The **show mac-address-table** command is also useful for monitoring general source learning activity and verifying dynamic VLAN assignments of addresses received on mobile ports.

1 Create VLAN 200, if it does not already exist, using the following command:

```
-> vlan 200
```

2 Assign switch ports 2 through 5 on slot 3 to VLAN 200--if they are not already associated with VLAN 200--using the following command:

```
-> vlan 200 port default 3/2-5
```

3 Create a static MAC address entry using the following command to assign address 000041:5BF30E to port 3/4 associated with VLAN 200 and to specify a timeout management status for the static address:

```
-> mac-address-table timeout 00:2d:95:5B:F3:0E 3/4 200
```

4 Change the MAC address aging time for VLAN 200 to 1200 seconds (the default is 300 seconds) using the following command:

```
-> mac-address-table aging-time 1200 vlan 200
```

Note. *Optional*. To verify the static MAC address configuration, enter **show mac-address-table**. For example:

```
-> show mac-address-table
Legend: Mac Address: * = address not valid

   Vlan      Mac Address          Type       Protocol    Operation    Interface
  ------+-------------------+--------------+-----------+------------+-----------
     1   00:00:00:00:00:01     learned        0800        bridging       8/1
     1   00:d0:95:6a:73:9a     learned        aaaa0003    bridging      10/23
   200   00:2d:95:5b:f3:0e     delontimeout   0           bridging       3/4

Total number of Valid MAC addresses above = 3
```

To verify the new aging time value for VLAN 200, enter **show mac-address-table aging-time vlan** followed by 200. For example,

```
-> show mac-address-table aging-time vlan 200

Mac Address Aging Time (seconds) for Vlan 200 = 1200
```

MAC Address Table Overview

Source learning builds and maintains the MAC address table on each switch. New MAC address table entries are created in one of two ways: they are dynamically learned or statically assigned. Dynamically learned MAC addresses are those that are obtained by the switch when source learning examines data packets and records the source address and the port and VLAN it was learned on. Static MAC addresses are user defined addresses that are statically assigned to a port and VLAN using the **mac-address-table** command.

Accessing MAC Address Table entries is useful for managing traffic flow and troubleshooting network device connectivity problems. For example, if a workstation connected to the switch is unable to communicate with another workstation connected to the same switch, the MAC address table might show that one of these devices was learned on a port that belonged to a different VLAN or the source MAC address of one of the devices may not appear at all in the address table.

Using Static MAC Addresses

Static MAC addresses are configured using the **mac-address-table** command. These addresses direct network traffic to a specific port and VLAN. They are particularly useful when dealing with silent network devices. These types of devices do not send packets, so their source MAC address is never learned and recorded in the MAC address table. Assigning a MAC address to the silent device's port creates a record in the MAC address table and ensures that packets destined for the silent device are forwarded out that port.

When defining a static MAC address for a particular slot/port and VLAN, consider the following:

- Configuring static MAC addresses is only supported on non-mobile ports.
- The specified slot/port must already belong to the specified VLAN. Use the **vlan port default** command to assign a port to a VLAN before you configure the static MAC address.
- Only traffic from other ports associated with the same VLAN is directed to the static MAC address slot/port.
- There are three types of static MAC addresses available: **permanent** (default), **reset**, or **timeout**. The type selected determines the status of the MAC address in the event of a switch reboot or when the MAC address age exceeds the aging timer. These types are defined as follows:

Status	Definition
permanent	MAC address remains in use even if MAC ages beyond the aging timer value or the switch is rebooted.
reset	MAC address is removed the next time the switch is rebooted.
timeout	MAC address is removed when it ages beyond the aging timer value.

  Note that static MAC addresses configured with a **reset** or **timeout** status are not captured when a snapshot of the switch's running configuration is taken.

- There are two types of static MAC address behavior supported: **bridging** (default) or **filtering**. Enter **filtering** to set up a denial of service to block potential hostile attacks. Traffic sent to or from a filtered MAC address is dropped. Enter **bridging** for regular traffic flow to or from the MAC address. For more information about Layer 2 filtering, see Chapter 24, "Configuring QoS."

- If a packet received on a port associated with the same VLAN contains a source address that matches a static MAC address, the packet is discarded. The packet's source address, however, is still learned on that port but is set to a filtering operational status. The same source address on different ports within the same VLAN is not supported.
- If a static MAC address is configured on a port link that is down or disabled, an asterisk appears to the right of the MAC address in the **show mac-address-table** command display. The asterisk indicates that this is an invalid MAC address. When the port link comes up, however, the MAC address is then considered valid and the asterisk no longer appears next to the address in the display.

Configuring Static MAC Addresses

To configure a permanent, bridging static MAC address on a VLAN port, enter **mac-address-table** followed by a MAC address, slot/port, and the VLAN ID to assign to the MAC address. For example, the following assigns a MAC address to port 10 on slot 4 associated with VLAN 255:

```
-> mac-address-table 00:02:DA:00:59:0C 4/10 255
```

Since **permanent** and **bridging** options for a static MAC are default settings, it is not necessary to enter them as part of the command.

The following configures a filtered static MAC address that source learning will remove from the MAC address table the next time the switch reboots:

```
-> mac-address-table reset 00:02:DA:00:59:0C 3/1 500 filtering
```

Use the **no** form of this command to clear MAC address entries from the table. If the MAC address status type (permanent, reset, or learned) is not specified, then only permanent addresses are removed from the table. The following example removes a MAC address entry with a reset status that is assigned on port 2 of slot 3 for VLAN 855 from the MAC address table:

```
-> no mac-address-table reset 00:00:02:CE:10:37 3/2 855
```

If a slot/port and VLAN ID are not specified when removing MAC address table entries, then all MACs defined with the specified status are removed. For example, the following command removes all learned MAC addresses from the table, regardless of their slot/port or VLAN assignments:

```
-> no mac-address-table learned
```

To verify static MAC address configuration and other table entries, use the **show mac-address-table** command. For more information about this command, see the *OmniSwitch CLI Reference Guide*.

Static MAC Addresses on Link Aggregate Ports

Static MAC Addresses are not assigned to physical ports that belong to a link aggregate. Instead, they are assigned to a link aggregate ID that represents a collection of physical ports. This ID is specified at the time the link aggregate of ports is created and when using the **mac-address-table** command.

To configure a permanent, bridging static MAC address on a link aggregate ID, enter **mac-address-table** followed by a MAC address, then **linkagg** followed by the link aggregate ID, and the VLAN ID to assign to the MAC address. For example, the following assigns a MAC address to link aggregate ID 2 associated with VLAN 455:

```
-> mac-address-table 00:95:2A:00:3E:4C linkagg 2 455
```

For more information about configuring a link aggregate of ports, see Chapter 11, "Configuring Static Link Aggregation," and Chapter 12, "Configuring Dynamic Link Aggregation."

Configuring MAC Address Table Aging Time

Source learning also tracks MAC address age and removes addresses from the MAC address table that have aged beyond the aging timer value. When a device stops sending packets, source learning keeps track of how much time has passed since the last packet was received on the device's switch port. When this amount of time exceeds the aging time value, the MAC is *aged out* of the MAC address table. Source learning always starts tracking MAC address age from the time since the last packet was received.

By default, the aging time is set to 300 seconds (5 minutes) and is configured on a per VLAN basis using the **mac-address-table aging-time** command. For example, the following sets the aging time for VLAN 255 to 1200 seconds (20 minutes):

```
-> mac-address-table aging-time 1200 vlan 255
```

A MAC address learned on a VLAN 255 port will age out if the time since a packet with that address was last seen on the port exceeds 1200 seconds. If a VLAN ID is not specified, then the aging time value is applied to all VLANs configured on the switch.

When using the **mac-address-table aging-time** command in a switch configuration file (e.g., **boot.cfg**), include an instance of this command specifying the VLAN ID for each VLAN configured on the switch. This is necessary even if all VLANs will have the same aging time value. If there is only one instance of this command in the configuration file and it does not specify a VLAN ID, the aging time value is applied only to VLAN 1.

Note. The MAC address table aging time is also used as the timeout value for the Address Resolution Protocol (ARP) table. This timeout value determines how long the switch retains dynamically learned ARP table entries. See Chapter 13, "Configuring IP," for more information.

To set the aging time back to the default value, use the **no** form of the **mac-address-table aging-time** command. For example, the following sets the aging time for VLAN 255 (for all VLANs if VLAN ID is not specified) back to the default of 300 seconds:

```
-> no mac-address-table aging-time vlan 255
```

To display the aging time value for one or all VLANs, use the **show mac-address-table aging-time** command. For more information about this command, see the *OmniSwitch CLI Reference Guide*.

Displaying MAC Address Table Information

To display MAC Address Table entries, statistics, and aging time values, use the show commands listed below:

show mac-address-table Displays a list of all MAC addresses known to the MAC address table, including static MAC addresses.

show mac-address-table count Displays a count of the different types of MAC addresses (learned, permanent, reset, and timeout). Also includes a total count of all addresses known to the MAC address table.

show mac-address-table aging-time Displays the current MAC address aging timer value by switch or VLAN.

For more information about the resulting displays from these commands, see the *OmniSwitch CLI Reference Guide*. An example of the output for the **show mac-address-table** and **show mac-address-table aging-time** commands is also given in "Sample MAC Address Table Configuration" on page 2-2.

3 Using 802.1s Multiple Spanning Tree

The Alcatel Multiple Spanning Tree (MST) implementation provides support for the IEEE 802.1s Multiple Spanning Tree Protocol (MSTP). In addition to the 802.1D Spanning Tree Algorithm and Protocol (STP) and the 802.1w Rapid Spanning Tree Algorithm and Protocol (RSTP), MSTP also ensures that there is always only one data path between any two switches for a given Spanning Tree instance to prevent network loops.

MSTP is an enhancement to the 802.1Q Common Spanning Tree (CST), which is provided when an Alcatel switch is running in the flat Spanning Tree operating mode. The flat mode applies a single spanning tree instance across all VLAN port connections on a switch. MSTP allows the configuration of Multiple Spanning Tree Instances (MSTIs) in addition to the CST instance. Each MSTI is mapped to a set of VLANs. As a result, flat mode can support the forwarding of VLAN traffic over separate data paths.

In addition to 802.1s MSTP support, the 802.1D STP and 802.1w RSTP are still available in either the flat or 1x1 mode. However, if using 802.1D or 802.1w in the flat mode, the single spanning tree instance per switch algorithm applies.

In This Chapter

This chapter describes 802.1s MST in general and how MSTP works on the switch. It provides information about configuring MSTP through the Command Line Interface (CLI). For more details about the syntax of commands, see the *OmniSwitch CLI Reference Guide*. For more information about Spanning Tree configuration commands as they apply to all supported protocols (STP, RSTP, and MSTP), see Chapter 6, “Configuring Spanning Tree Parameters.”

The following topics are included in this chapter as they relate to the Alcatel implementation of the 802.1s MSTP standard:

- “MST General Overview” on page 3-4.
- “MST Configuration Overview” on page 3-10.
- “Using Spanning Tree Configuration Commands” on page 3-10.
- “MST Interoperability and Migration” on page 3-12.
- “Quick Steps for Configuring an MST Region” on page 3-14.
- “Quick Steps for Configuring MSTIs” on page 3-16.
- “Verifying the MST Configuration” on page 3-19.

MST Specifications

IEEE Standards supported	802.1D–*Media Access Control (MAC) Bridges* 802.1w–*Rapid Reconfiguration (802.1D Amendment 2)* 802.1Q–*Virtual Bridged Local Area Networks* 802.1s–*Multiple Spanning Trees (802.1Q Amendment 3)*
Spanning Tree Operating Modes supported	Flat mode - one spanning tree instance per switch 1x1 mode - one spanning tree instance per VLAN
Spanning Tree Protocols supported	802.1D Standard Spanning Tree Algorithm and Protocol (STP) 802.1w Rapid Spanning Tree Algorithm and Protocol (RSTP) 802.1s Multiple Spanning Tree Algorithm and Protocol (MSTP)
Spanning Tree port eligibility	Fixed ports (non-mobile) 802.1Q tagged ports Link aggregate of ports
Maximum 1x1 mode Spanning Tree instances per switch	253
Maximum flat mode 802.1s Multiple Spanning Tree Instances (MSTI) per switch	16 MSTI, in addition to the Common and Internal Spanning Tree instance (also referred to as MSTI 0).
CLI Command Prefix Recognition	All Spanning Tree commands support prefix recognition. See the “Using the CLI” chapter in the *OmniSwitch 7700/7800/8800 Switch Management Guide* for more information.

Spanning Tree Bridge Parameter Defaults

Parameter Description	Command	Default
Spanning Tree operating mode	**bridge mode**	1x1 (a separate Spanning Tree instance for each VLAN)
Spanning Tree protocol	**bridge protocol**	STP (802.1D)
BPDU switching status.	**bridge bpdu-switching**	Disabled
Priority value for the Spanning Tree instance.	**bridge priority**	32768
Hello time interval between each BPDU transmission.	**bridge hello time**	2 seconds
Maximum aging time allowed for Spanning Tree information learned from the network.	**bridge max age**	20 seconds
Spanning Tree port state transition time.	**bridge forward delay**	15 seconds

Spanning Tree Port Parameter Defaults

Parameter Description	Command	Default
Spanning Tree port administrative state	**bridge slot/port**	Enabled
Spanning Tree port priority value	**bridge slot/port priority**	7
Spanning Tree port path cost.	**bridge slot/port path cost**	0 (cost is based on port speed)
Path cost mode	**bridge path cost mode**	AUTO (16-bit in 1x1 mode, 32-bit in flat mode)
Port state management mode	**bridge slot/port mode**	Dynamic (Spanning Tree Algorithm determines port state)
Type of port connection	**bridge slot/port connection**	auto point to point

MST Region Defaults

Although the following parameter values are specific to the MSTP (802.1s), they are configurable regardless of which mode (flat or 1x1) or protocol is active on the switch.

Parameter Description	Command	Default
The MST region name	**bridge mst region name**	blank
The revision level for the MST region	**bridge mst region revision level**	0
The maximum number of hops authorized for the region	**bridge mst region max hops**	20
The number of Multiple Spanning Tree Instances (MSTI).	**bridge msti**	1 (flat mode instance)
The VLAN to MSTI mapping.	**bridge msti vlan**	All VLANs are mapped to the Common Internal Spanning Tree (CIST) instance

MST General Overview

The Multiple Spanning Tree (MST) feature allows for the mapping of one or more VLANs to a single Spanning Tree instance, referred to as a Multiple Spanning Tree Instance (MSTI), when the switch is running in the flat Spanning Tree mode. MST uses the Multiple Spanning Tree Algorithm and Protocol (MSTP) to define the Spanning Tree path for each MSTI. In addition, MSTP provides the ability to group switches into MST Regions. An MST Region appears as a single, flat Spanning Tree instance to switches outside the region.

This section provides an overview of the MST feature that includes the following topics:

- "How MSTP Works" on page 3-4.
- "Comparing MSTP with STP and RSTP" on page 3-7
- "What is a Multiple Spanning Tree Instance (MSTI)" on page 3-7.
- "What is a Multiple Spanning Tree Region" on page 3-8.
- "What is the Internal Spanning Tree (IST) Instance" on page 3-9.
- "What is the Common and Internal Spanning Tree Instance" on page 3-9.

How MSTP Works

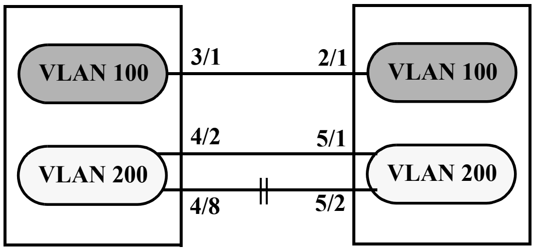
MSTP, as defined in the IEEE 802.1s standard, is an enhancement to the IEEE 802.1Q Common Spanning Tree (CST). The CST is a single spanning tree that uses 802.1D (STP) or 802.1w (RSTP) to provide a loop-free network topology.

The Alcatel flat spanning tree mode applies a single CST instance on a per switch basis. The 1x1 mode is an Alcatel proprietary implementation that applies a single spanning tree instance on a per VLAN basis. MSTP is only supported in the flat mode and allows for the configuration of additional spanning tree instances instead of just the one CST.

On Alcatel 802.1s flat mode switches, the CST is represented by the Common and Internal Spanning Tree (CIST) instance 0 and exists on all switches. Up to 17 instances, including the CIST, are supported. Each additional instance created is referred to as a Multiple Spanning Tree Instance (MSTI). An MSTI represents a configurable association between a single Spanning Tree instance and a set of VLANs.

Note that although MSTP provides the ability to define MSTIs while running in the flat mode, port state and role computations are still automatically calculated by the CST algorithm across all MSTIs. However, it is possible to configure the priority and/or path cost of a port for a particular MSTI so that a port remains in a forwarding state for an MSTI instance, even if it is blocked as a result of automatic CST computations for other instances.

The following diagrams help to further explain how MSTP works by comparing how port states are determined on 1x1 STP/RSTP mode, flat mode STP/RSTP, and flat mode MSTP switches.

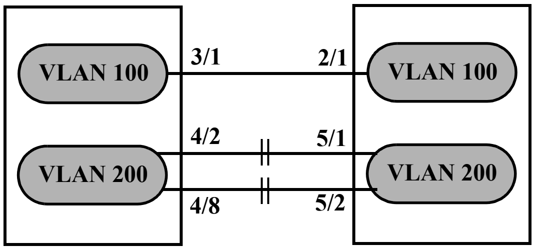


1x1 Mode STP/RSTP

In the above 1x1 mode example:

- Both switches are running in the 1x1 mode (one Spanning Tree instance per VLAN).
- VLAN 100 and VLAN 200 are each associated with their own Spanning Tree instance.
- The connection between 3/1 and 2/1 is left in a forwarding state because it is part of the VLAN 100 Spanning Tree instance and is the only connection for that instance.

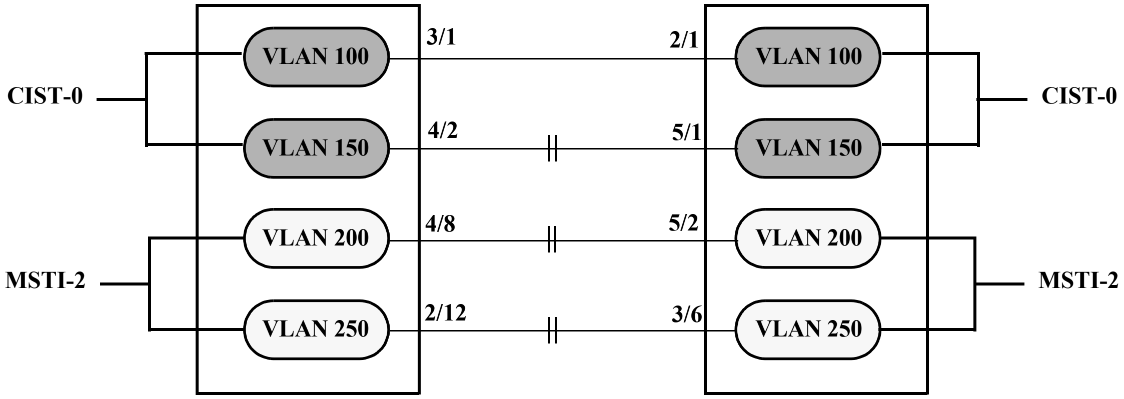
  Note that if additional switches containing a VLAN 100 were attached to the switches in this diagram, the 3/1 to 2/1 connection could also go into blocking if the VLAN 100 Spanning Tree instance determines it is necessary to avoid a network loop.
- The connections between 4/8 and 5/2 and 4/2 and 5/1 are seen as redundant because they are both controlled by the VLAN 200 Spanning Tree instance and connect to the same switches. The VLAN 200 Spanning Tree instance determines which connection provides the best data path and transitions the other connection to a blocking state.



Flat Mode STP/RSTP (802.1D/802.1w)

In the above flat mode STP/RSTP example:

- Both switches are running in the flat mode. As a result, a single flat mode Spanning Tree instance applies to the entire switch and compares port connections across VLANs to determine which connection provides the best data path.
- The connection between 3/1 and 2/1 is left forwarding because the flat mode instance determined that this connection provides the best data path between the two switches.
- The 4/8 to 5/2 connection and the 4/2 to 5/1 connection are considered redundant connections so they are both blocked in favor of the 3/1 to 2/1 connection.



Flat Mode MSTP (802.1s)

In the above flat mode MSTP example:

- Both switches are running in the flat mode and using MSTP.
- VLANs 100 and 150 are *not* associated with an MSTI. By default they are controlled by the CIST instance 0, which exists on every switch.
- VLANs 200 and 250 are associated with MSTI 2 so their traffic can traverse a path different from that determined by the CIST.
- Ports are blocked the same way they were blocked in the flat mode STP/RSTP example; all port connections are compared to each other across VLANs to determine which connection provides the best path.

  However, because VLANs 200 and 250 are associated to MSTI 2, it is possible to change the port path cost for ports 2/12, 3/6, 4/8 and/or 5/2 so that they provide the best path for MSTI 2 VLANs, but do not carry CIST VLAN traffic or cause CIST ports to transition to a blocking state.

  Another alternative is to assign all VLANs to an MSTI, leaving no VLANs controlled by the CIST. As a result, the CIST BPDU will only contain MSTI information.

See "Quick Steps for Configuring MSTIs" on page 3-16 for more information about how to direct VLAN traffic over separate data paths using MSTP.

Comparing MSTP with STP and RSTP

Using MSTP (802.1s) has the following items in common with STP (802.1D) and RSTP (802.1w) protocols:

- Each protocol ensures one data path between any two switches within the network topology. This prevents network loops from occurring while at the same time allowing for redundant path configuration.
- Each protocol provides automatic reconfiguration of the network Spanning Tree topology in the event of a connection failure and/or when a switch is added to or removed from the network.
- All three protocols are supported in the flat Spanning Tree operating mode.
- The flat mode CST instance automatically determines port states and roles across VLAN port and MSTI associations. This is because the CST instance is active on all ports and only one BPDU is used to forward information for all MSTIs.
- MSTP is based on RSTP.

Using MSTP differs from STP and RSTP as follows:

- MSTP is only supported when the switch is running in the flat Spanning Tree mode. STP and RSTP are supported in both the 1x1 and flat modes.
- MSTP allows for the configuration of up to 16 Multiple Spanning Tree Instances (MSTI) in addition to the CST instance. Flat mode STP and RSTP protocols only use the single CST instance for the entire switch. See "What is a Multiple Spanning Tree Instance (MSTI)" on page 3-7 for more information.
- MSTP applies a single Spanning Tree instance to an MSTI ID number that represents a set of VLANs; a one to many association. STP and RSTP in the flat mode apply one Spanning Tree instance to all VLANs; a one to all association. STP and RSTP in the 1x1 mode apply a single Spanning Tree instance to each existing VLAN; a one to one association.
- The port priority and path cost parameters are configurable for an individual MSTI that represents the VLAN associated with the port.
- The flat mode 802.1D or 802.1w CST is identified as instance 1. When using MSTP, the CST is identified as CIST (Common and Internal Spanning Tree) instance 0. See "What is the Common and Internal Spanning Tree Instance" on page 3-9 for more information.
- MSTP allows the segmentation of switches within the network into MST regions. Each region is seen as a single virtual bridge to the rest of the network, even though multiple switches may belong to the one region. See "What is a Multiple Spanning Tree Region" on page 3-8 for more information.
- MSTP has lower overhead than a 1x1 configuration. In 1x1 mode, because each VLAN is assigned a separate Spanning Tree instance, BPDUs are forwarded on the network for each VLAN. MSTP only forwards one BPDU for the CST that contains information for all configured MSTI on the switch.

What is a Multiple Spanning Tree Instance (MSTI)

An MSTI is a single Spanning Tree instance that represents a group of VLANs. Alcatel switches support up to 16 MSTIs on one switch. This number is in addition to the Common and Internal Spanning Tree (CIST) instance 0, which is also known as MSTI 0. The CIST instance exists on every switch. By default, all VLANs not mapped to an MSTI are associated with the CIST instance. See "What is the Common and Internal Spanning Tree Instance" on page 3-9 for more information.

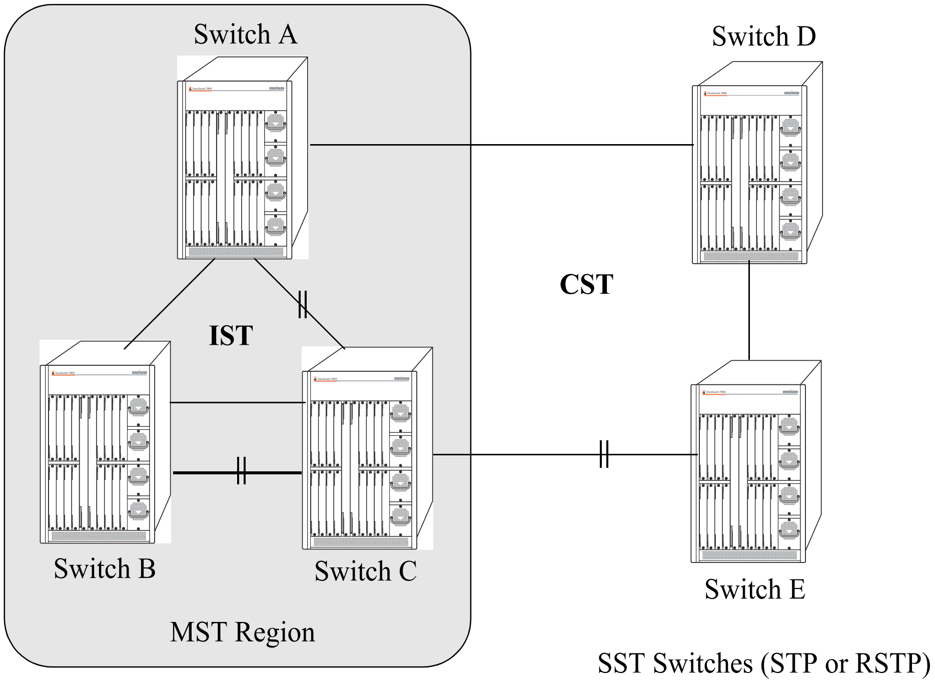
What is a Multiple Spanning Tree Region

A Multiple Spanning Tree region represents a group of 802.1s switches. An MST region appears as a single, flat mode instance to switches outside the region. A switch can belong to only one region at a time. The region a switch belongs to is identified by the following configurable attributes, as defined by the IEEE 802.1s standard:

- **Region name**–An alphanumeric string up to 32 characters.
- **Region revision level**–A numerical value between 0 and 65535.
- **VLAN to MSTI table**–Generated when VLANs are associated with MSTIs. Identifies the VLAN to MSTI mapping for the switch.

Switches that share the same values for the configuration attributes described above belong to the same region. For example, in the diagram below:

- Switches A, B, and C all belong to the same region because they all are configured with the same region name, revision level, and have the same VLANs mapped to the same MSTI.
- The CST for the entire network sees Switches A, B, and C as one virtual bridge that is running a single Spanning Tree instance. As a result, CST blocks the path between Switch C and Switch E instead of blocking a path between the MST region switches to avoid a network loop.
- The paths between Switch A and Switch C and the redundant path between Switch B and Switch C were blocked as a result of the Internal Spanning Tree (IST) computations for the MST Region. See "What is the Internal Spanning Tree (IST) Instance" on page 3-9 for more information.



In addition to the attributes described above, the MST maximum hops parameter defines the number of bridges authorized to propagate MST BPDU information. In essence, this value defines the size of the region in that once the maximum number of hops is reached, the BPDU is discarded. The maximum number of hops for the region, however, is not one of the attributes that defines whether or not a switch is a member of a region.

See "Quick Steps for Configuring an MST Region" on page 3-14 for a tutorial on how to configure MST region parameters.

What is the Common Spanning Tree

The Common Spanning Tree (CST) is the overall network Spanning Tree topology resulting from STP, RSTP, and/or MSTP calculations to provide a single data path through the network. CST provides connectivity between MST regions and other MST regions and/or Single Spanning Tree (SST) switches. For example, in the above diagram, CST calculations detected a network loop created by the connections between Switch D, Switch E, and the MST Region. As a result, one of the paths was blocked.

What is the Internal Spanning Tree (IST) Instance

The IST instance determines and maintains the CST topology between MST switches that belong to the same MST region. In other words, the IST is simply a CST that only applies to MST Region switches while at the same time representing the region as a single Spanning Tree bridge to the network CST.

As shown in the above diagram, the redundant path between Switch B and Switch C is blocked and the path between Switch A and Switch C is blocked. These blocking decisions were based on IST computations within the MST region. IST sends and receives BPDU to/from the network CST. MSTI within the region do not communicate with the network CST. As a result, the CST only sees the IST BPDU and treats the MST region as a single Spanning Tree bridge.

What is the Common and Internal Spanning Tree Instance

The Common and Internal Spanning Tree (CIST) instance is the Spanning Tree calculated by the MST region IST and the network CST. The CIST is represented by the single Spanning Tree flat mode instance that is available on all switches. By default, all VLANs are associated to the CIST until they are mapped to an MSTI.

When using STP (802.1D) or RSTP (802.1w), the CIST is also known as instance 1 or bridge 1. When using MSTP (802.1s), the CIST is also known as instance 0 or MSTI 0.

Note that when MSTP (802.1s) is the active flat mode protocol, explicit Spanning Tree bridge commands are required to configure parameter values. Implicit commands are for configuring parameters when the STP or RSTP protocols are in use. See "Using Spanning Tree Configuration Commands" on page 3-10 for more information.

MST Configuration Overview

The following general steps are required to set up a Multiple Spanning Tree (MST) configuration:

- **Select the flat Spanning Tree mode.** By default, each switch runs in the 1x1 mode. MSTP is only supported on a flat mode switch. See "Understanding Spanning Tree Modes" on page 3-11 for more information.
- **Select the 802.1s protocol**. By default, each switch uses the 802.1D protocol. Selecting 802.1s activates the Multiple Spanning Tree Protocol (MSTP). See "How MSTP Works" on page 3-4 for more information.
- **Configure an MST region name and revision level.** Switches that share the same MST region name, revision level, and VLAN to Multiple Spanning Tree Instance (MSTI) mapping belong to the same MST region. See "What is a Multiple Spanning Tree Region" on page 3-8 for more information.
- **Configure MSTIs**. By default, every switch has a Common and Internal Spanning Tree (CIST) instance 0, which is also referred to as MSTI 0. Configuration of additional MSTI is required to segment switch VLANs into separate instances. See "What is a Multiple Spanning Tree Instance (MSTI)" on page 3-7 for more information.
- **Map VLANs to MSTI.** By default, all existing VLANs are mapped to the CIST instance 0. Associating a VLAN to an MSTI specifies which Spanning Tree instance will determine the best data path for traffic carried on the VLAN. In addition, the VLAN-to-MSTI mapping is also one of three MST configuration attributes used to determine that the switch belongs to a particular MST region.

For a tutorial on setting up an example MST configuration, see "Quick Steps for Configuring an MST Region" on page 3-14 and "Quick Steps for Configuring MSTIs" on page 3-16.

Using Spanning Tree Configuration Commands

The Alcatel implementation of the 802.1s Multiple Spanning Tree Protocol introduces the concept of *implicit* and *explicit* CLI commands for Spanning Tree configuration and verification. Explicit commands contain one of the following keywords that specifies the type of Spanning Tree instance to modify:

- **cist**–command applies to the Common and Internal Spanning Tree instance.
- **msti**–command applies to the specified 802.1s Multiple Spanning Tree Instance.
- **1x1**–command applies to the specified VLAN instance.

Explicit commands allow the configuration of a particular Spanning Tree instance independent of which mode and/or protocol is currently active on the switch. The configuration, however, does not go active until the switch is changed to the appropriate mode. For example, if the switch is running in the 1x1 mode, the following explicit commands changes the MSTI 3 priority to 12288:

```
-> bridge msti 3 priority 12288
```

Even though the above command is accepted in the 1x1 mode, the new priority value does not take effect until the switch mode is changed to flat mode.

Note that explicit commands using the **cist** and **msti** keywords are required to define an MSTP (802.1s) configuration. Implicit commands are only allowed for defining STP or RSTP configurations.

Implicit commands resemble previously implemented Spanning Tree commands, but apply to the appropriate instance based on the current mode and protocol that is active on the switch. For example, if the 1x1 mode is active, the instance number specified with the following command implies a VLAN ID:

```
-> bridge 255 priority 16384
```

If the flat mode is active, the single flat mode instance is implied and thus configured by the command. Since the flat mode instance is implied in this case, there is no need to specify an instance number. For example, the following command configures the protocol for the flat mode instance:

```
-> bridge protocol mstp
```

Similar to previous releases, it is possible to configure the flat mode instance by specifying **1** for the instance number (e.g., **bridge 1 protocol rstp**). However, this is only available when the switch is already running in the flat mode and STP or RSTP is the active protocol.

Note. When a snapshot is taken of the switch configuration, the explicit form of all Spanning Tree commands is captured. For example, if the priority of MSTI 2 was changed from the default value to a priority of 16384, then **bridge msti 2 priority 16384** is the command captured to reflect this in the snapshot file. In addition, explicit commands are captured for both flat and 1x1 mode configurations.

For more information about Spanning Tree configuration commands as they apply to all supported protocols (STP, RSTP, and MSTP), see Chapter 6, "Configuring Spanning Tree Parameters."

Understanding Spanning Tree Modes

The switch can operate in one of two Spanning Tree modes: *flat* and *1x1*. The flat mode provides a Common Spanning Tree (CST) instance that applies across all VLANs by default. This mode supports the use of the STP (802.1D), RSTP (802.1w), and MSTP (802.1s) protocols. MSTP allows the mapping of one or more VLANs to a single Spanning Tree instance.

The 1x1 mode is an Alcatel proprietary implementation that automatically calculates a separate Spanning Tree instance for each VLAN configured on the switch. This mode only supports the use of the STP and RSTP protocols.

Although MSTP is not supported in the 1x1 mode, it is possible to define an MSTP configuration in this mode using explicit Spanning Tree commands. See "Using Spanning Tree Configuration Commands" on page 3-10 for more information about explicit commands.

By default, a switch is running in the 1x1 mode and using the 802.1D protocol when it is first turned on. See Chapter 6, "Configuring Spanning Tree Parameters," for more information about Spanning Tree modes.

MST Interoperability and Migration

Connecting an MSTP (802.1s) switch to a non-MSTP flat mode switch is supported. Since the Common and Internal Spanning Tree (CIST) controls the flat mode instance on both switches, STP or RSTP can remain active on the non-MSTP switch within the network topology.

An MSTP switch is part of a Multiple Spanning Tree (MST) Region, which appears as a single, flat mode instance to the non-MSTP switch. The port that connects the MSTP switch to the non-MSTP switch is referred to as a *boundary* port. When a boundary port detects an STP (802.1D) or RSTP (802.1w) BPDU, it responds with the appropriate protocol BPDU to provide interoperability between the two switches. This interoperability also serves to indicate the edge of the MST region.

Interoperability between 802.1s MSTP switches and 1x1 mode switches is not recommended. The 1x1 mode is a proprietary implementation that creates a separate Spanning Tree instance for each VLAN configured on the switch. The 802.1s MSTP implementation is in compliance with the IEEE standard and is only supported on flat mode switches.

Tagged BPDU transmitted from a 1x1 switch are ignored by a flat mode switch, which can cause a network loop to go undetected. Although it is not recommended, it may be necessary to temporarily connect a 1x1 switch to a flat mode switch until migration to MSTP is complete. If this is the case, then only configure a fixed, untagged connection between VLAN 1 on both switches.

Migrating from Flat Mode STP/RSTP to Flat Mode MSTP

Migrating an STP/RSTP flat mode switch to MSTP is relatively transparent. When STP or RSTP is the active protocol, the Common and Internal Spanning Tree (CIST) controls the flat mode instance. If on the same switch the protocol is changed to MSTP, the CIST still controls the flat mode instance.

Note the following when converting a flat mode STP/RSTP switch to MSTP:

- Making a backup copy of the switch **boot.cfg** file before changing the protocol to MSTP is highly recommended. Having a backup copy will make it easier to revert to the non-MSTP configuration if necessary. Once MSTP is active, commands are written in their explicit form and not compatible with previous releases of Spanning Tree.

- When converting multiple switches, change the protocol to MSTP first on every switch before starting to configure Multiple Spanning Tree Instances (MSTI).

- Once the protocol is changed, MSTP features are available for configuration. Multiple Spanning Tree Instances (MSTI) are now configurable for defining data paths for VLAN traffic. See “How MSTP Works” on page 3-4 for more information.

- Using explicit Spanning Tree commands to define the MSTP configuration is required. Implicit commands are for configuring STP and RSTP. See “Using Spanning Tree Configuration Commands” on page 3-10 for more information.

- STP and RSTP use a 16-bit port path cost (PPC) and MSTP uses a 32-bit PPC. When the protocol is changed to MSTP, the bridge priority and PPC values for the flat mode CIST instance are reset to their default values.

- It is possible to configure the switch to use 32-bit PPC value for all protocols (see the **bridge path cost mode** command page for more information). If this is the case, then the PPC for the CIST is not reset when the protocol is changed to/from MSTP.

- This implementation of MSTP is compliant with the IEEE 802.1s standard and thus provides interconnectivity with 802.1s compliant systems.

Migrating from 1x1 Mode to Flat Mode MSTP

As previously described, the 1x1 mode is an Alcatel proprietary implementation that applies one Spanning Tree instance to each VLAN. For example, if five VLANs exist on the switch, then their are five Spanning Tree instances active on the switch, unless Spanning Tree is disabled on one of the VLANs.

Note the following when converting a 1x1 mode STP/RSTP switch to flat mode MSTP:

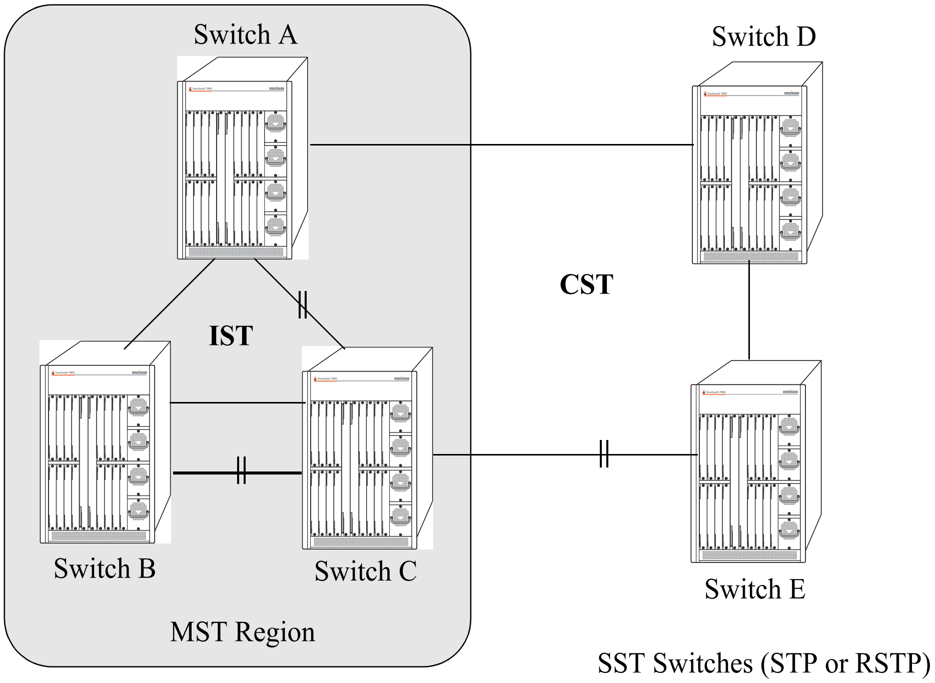
- Making a backup copy of the switch **boot.cfg** file before changing the protocol to MSTP is highly recommended. Having a backup copy will make it easier to revert to the non-MSTP configuration if necessary. Once MSTP is active, commands are written in their explicit form and not compatible with previous releases of Spanning Tree. If the need arises
- Using MSTP requires changing the switch mode from 1x1 to flat. When the mode is changed from 1x1 to flat, ports still retain their VLAN associations but are now part of a single, flat mode Spanning Tree instance that spans across all VLANs. As a result, a path that was forwarding traffic in the 1x1 mode may transition to a blocking state after the mode is changed to flat.
- Once the protocol is changed, MSTP features are available for configuration. Multiple Spanning Tree Instances (MSTI) are now configurable for defining data paths for VLAN traffic. See "How MSTP Works" on page 3-4 for more information.
- Note that STP/RSTP use a 16-bit port path cost (PPC) and MSTP uses a 32-bit PPC. When the protocol is changed to MSTP, the bridge priority and PPC values for the flat mode CIST instance are reset to their default values.
- It is possible to configure the switch to use 32-bit PPC value for all protocols (see the **bridge path cost mode** command page for more information). If this is the case, then the PPC for the CIST is not reset when the protocol is changed to/from MSTP.
- This implementation of MSTP is compliant with the IEEE 802.1s standard and thus provides interconnectivity with 802.1s compliant systems.

Quick Steps for Configuring an MST Region

An MST region identifies a group of MSTP (802.1s) switches that is seen as a single, flat mode instance by other regions and/or non-MSTP switches. A region is defined by three attributes: name, revision level, and a VLAN-to-MSTI mapping. Switches configured with the same value for all three of these attributes belong to the same MST region.

Note that an additional configurable MST region parameter defines the maximum number of hops authorized for the region but is not considered when determining regional membership.The maximum hops value is the value used by all bridges within the region when the bridge is acting as the root of the MST region.

This section provides a tutorial for defining a sample MST region configuration, as shown in the diagram below.



In order for switches A, B, and C in the above diagram to belong to the same MST region, they must all share the same values for region name, revision level, and configuration digest (VLAN-to-MSTI mapping).

The following steps are performed on each switch to define **Alcatel Marketing** as the MST region name, **2000** as the MST region revision level, map exiting VLANs to existing MSTIs, and **3** as the maximum hops value for the region:

1 Configure an MST Region name using the **bridge mst region name** command. For example:

```
-> bridge mst region name “Alcatel Marketing”
```

2 Configure the MST Region revision level using the **bridge mst region revision level** command. For example:

```
-> bridge mst region revision level 2000
```

3 Map VLANs 100 and 200 to MSTI 2 and VLANs 300 and 400 to MSTI 4 using the **bridge msti vlan** command to define the configuration digest. For example:

```
-> bridge msti 2 vlan 100 200
-> bridge msti 4 vlan 300 400
```

See “Quick Steps for Configuring MSTIs” on page 3-16 for a tutorial on how to create and map MSTIs to VLANs.

4 Configure **3** as the maximum number of hops for the region using the **bridge mst region max hops** command. For example:

```
-> bridge mst region max hops 3
```

Note. (*Optional*) Verify the MST region configuration on each switch with the **show spantree mst region** command. For example:

```
-> show spantree mst region
Configuration Name        : Alcatel Marketing,
Revision Level            : 2000,
Configuration Digest      : 0x922fb3f 31752d68 67fe1155 d0ce8380,
Revision Max hops         : 3,
Cist Instance Number       : 0
```

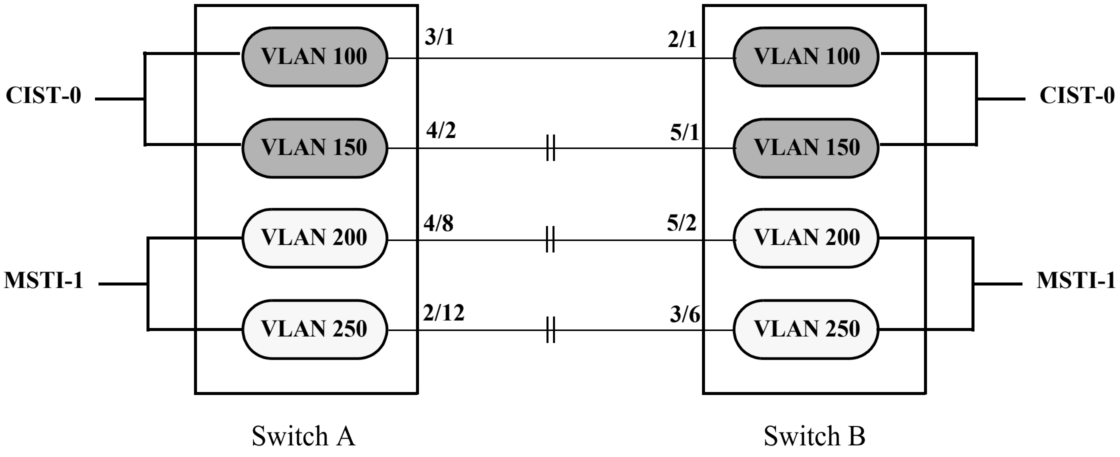
All switches configured with the exact same values as shown in the above example are considered members of the Alcatel Marketing MST region.

Quick Steps for Configuring MSTIs

By default the Spanning Tree software is active on all switches and operating in the 1x1 mode using the standard 802.1D STP. As a result, a loop-free network topology is automatically calculated based on default 802.1D Spanning Tree switch, bridge, and port parameter values.

Using Multiple Spanning Tree (MST) requires configuration changes to the default Spanning Tree values (mode and protocol) as well as defining specific MSTP parameters and instances.

The following steps provide a tutorial for setting up a sample MSTP configuration, as shown in the diagram below:



Flat Mode MSTP (802.1s) Quick Steps Example

1 Change the Spanning Tree operating mode, if necessary, on Switch A and Switch B from 1x1 to flat mode using the **bridge mode** command. For example:

```
-> bridge mode flat
```

Note that defining an MSTP configuration requires the use of explicit Spanning Tree commands, which are available in both the flat and 1x1 mode. As a result, this step is optional. See "Using Spanning Tree Configuration Commands" on page 3-10 for more information.

2 Change the Spanning Tree protocol to 802.1s using the **bridge protocol** command. For example:

```
-> bridge protocol mstp
```

3 Create VLANs 100, 200, 300, and 400 using the **vlan** command. For example:

```
-> vlan 100
-> vlan 150
-> vlan 200
-> vlan 250
```

4 Assign switch ports to VLANs, as shown in the above diagram, using the **vlan port default** command. For example, the following commands assign ports 3/1, 4/2, 4/8, and 2/12 to VLANs 100, 150, 200, and 250 on Switch A:

```
-> vlan 100 port default 3/1
-> vlan 150 port default 4/2
-> vlan 200 port default 4/8
-> vlan 250 port default 2/12
```

The following commands assign ports 2/1, 5/1, 5/2, and 3/6 to VLANs 100, 150, 200, and 250 on Switch B:

```
-> vlan 100 port default 2/1
-> vlan 150 port default 5/1
-> vlan 200 port default 5/2
-> vlan 250 port default 3/6
```

5 Create one MSTI using the **bridge msti** command. For example:

```
-> bridge msti 1
```

6 Assign VLANs 200 and 250 to MSTI 1. For example:

```
-> bridge msti 1 vlan 100 200
```

By default, all VLANs are associated with the CIST instance. As a result, VLANs 100 and 150 do not require any configuration to map them to the CIST instance.

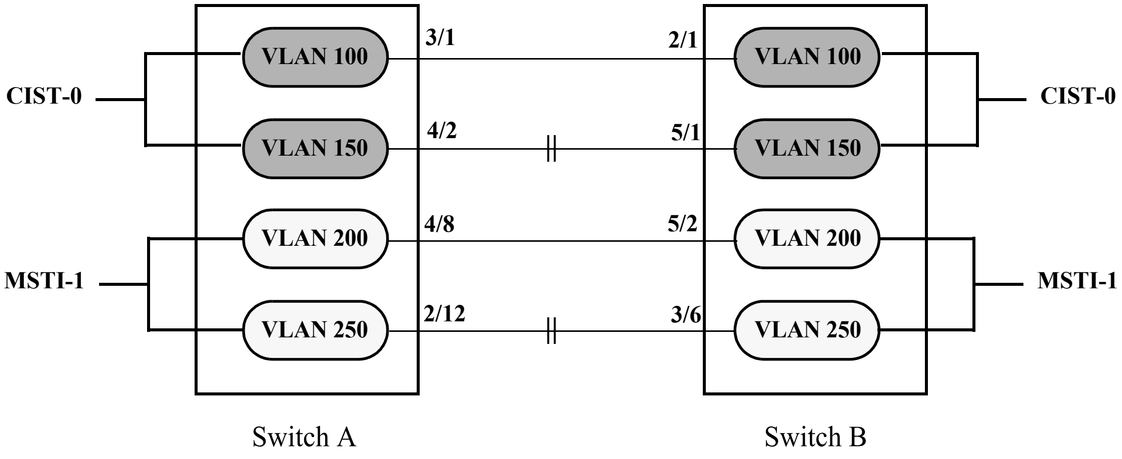
7 Configure the port path cost (PPC) for all ports on both switches associated with MSTI 1 to a PPC value that is lower than the PPC value for the ports associated with the CIST instance using the **bridge msti slot/port path cost** command. For example, the PPC for ports associated with the CIST instance is set to the default of 200,000 for 100 MB connections. The following commands change the PPC value for ports associated with the MSTI 1 to 20,000:

```
-> bridge msti 1 4/8 path cost 20,000
-> bridge msti 1 2/12 path cost 20,000
-> bridge msti 1 5/2 path cost 20,000
-> bridge msti 1 3/6 path cost 20,000
```

Note that in this example, port connections between VLANs 150, 200, and 250 on each switch initially were blocked, as shown in the diagram on page 3-16. This is because in flat mode MSTP, each instance is active on all ports resulting in a comparison of connections independent of VLAN and MSTI associations.

To avoid this and allow VLAN traffic to flow over separate data paths based on MSTI association, Step 7 of this tutorial configures a superior port path cost value for ports associated with MSTI 1. As a result, MSTI 1 selects one of the data paths between its VLANs as the best path, rather than the CIST data paths, as shown in the diagram on page 3-17.

:



Flat Mode MSTP (802.1s) with Superior MSTI 1 PPC Values

Note that of the two data paths available to MSTI 1 VLANs, one is still blocked because it is seen as redundant for that instance. In addition, the CIST data path still remains available for CIST VLAN traffic.

Another solution to this scenario is to assign all VLANs to an MSTI, leaving no VLANs controlled by the CIST. As a result, the CIST BPDU will only contain MSTI information. See "How MSTP Works" on page 3-4 for more information.

Verifying the MST Configuration

To display information about the MST configuration on the switch, use the show commands listed below:

show spantree cist Displays the Spanning Tree bridge configuration for the flat mode Common and Internal Spanning Tree (CIST) instance.

show spantree msti Displays Spanning Tree bridge information for an 802.1s Multiple Spanning Tree Instance (MSTI).

show spantree cist ports Displays Spanning Tree port information for the flat mode Common and Internal Spanning Tree (CIST) instance.

show spantree msti ports Displays Spanning Tree port information for a flat mode 802.1s Multiple Spanning Tree Instance (MSTI).

show spantree mst region Displays the Multiple Spanning Tree (MST) region information for the switch.

show spantree cist vlan-map Displays the range of VLANs associated with the flat mode Common and Internal Spanning Tree (CIST) instance.

show spantree msti vlan-map Displays the range of VLANs associated with the specified Multiple Spanning Tree Instance (MSTI).

show spantree map-msti Displays the Multiple Spanning Tree Instance (MSTI) that is associated to the specified VLAN.

show spantree mst port Displays a summary of Spanning Tree connection information and instance associations for the specified port or a link aggregate of ports.

For more information about the resulting displays from these commands, see the *OmniSwitch CLI Reference Guide*.

4 Configuring Learned Port Security

Learned Port Security (LPS) provides a mechanism for authorizing source learning of MAC addresses on Ethernet and Gigabit Ethernet ports. The only types of Ethernet ports that LPS does not support are link aggregate and tagged (trunked) link aggregate ports. Using LPS to control source MAC address learning provides the following benefits:

- A configurable source learning time limit that applies to all LPS ports.
- A configurable limit on the number of MAC addresses allowed on an LPS port.
- Dynamic configuration of a list of authorized source MAC addresses.
- Static configuration of a list of authorized source MAC addresses.
- Two methods for handling unauthorized traffic: stopping all traffic on the port or only blocking traffic that violates LPS criteria.

In This Chapter

This chapter describes how to configure LPS parameters through the Command Line Interface (CLI). CLI commands are used in the configuration examples; for more details about the syntax of commands, see the *OmniSwitch CLI Reference Guide*.

Configuration procedures described in this chapter include:

- Enabling LPS for a port on page 4-7.
- Specifying a source learning time limit for all LPS ports on page 4-7.
- Configuring the maximum number of MAC addressees learned per port on page 4-8.
- Configuring a list of authorized MAC addresses for an LPS port on page 4-8.
- Configuring a range of authorized MAC addresses for an LPS port on page 4-9.
- Selecting the security violation mode for an LPS port on page 4-10.
- Displaying LPS configuration information on page 4-11.

For more information about source MAC address learning, see Chapter 2, “Managing Source Learning.”

Learned Port Security Specifications

RFCs supported	Not applicable at this time.
IEEE Standards supported	Not applicable at this time.
Ports eligible for Learned Port Security	Ethernet and Gigabit Ethernet ports (fixed, mobile, 802.1Q tagged, and authenticated ports).
Ports not eligible for Learned Port Security	Link aggregate ports. 802.1Q (trunked) link aggregate ports.
Minimum number of learned MAC addresses allowed per port	1
Maximum number of learned MAC addresses allowed per port	100
Maximum number of configurable MAC address ranges per LPS port.	1
Maximum number of learned MAC addresses per network interface (NI) module (applies to all ports on the module.	32K
Maximum number of learned MAC addresses per switch (applies to all switch ports).	64K

Learned Port Security Defaults

Parameter Description	Command	Default
LPS status for a port.	**port-security**	disabled
Number of learned MAC addresses allowed on an LPS port.	**port security maximum**	1
Source learning time limit.	**port-security shutdown**	disabled
Configured MAC addresses per LPS port.	**port-security mac**	none
MAC address range per LPS port.	**port-security mac-range**	00:00:00:00:00:00–ff:ff:ff:ff:ff:ff
LPS port violation mode.	**port-security violation**	restrict

Sample Learned Port Security Configuration

This section provides a quick tutorial that demonstrates the following tasks:

- Enabling LPS on a set of switch ports.
- Defining the maximum number of learned MAC addresses allowed on an LPS port.
- Defining the time limit in which source learning is allowed on all LPS ports.
- Selecting a method for handling unauthorized traffic received on an LPS port.

Note that LPS is supported on 10/100 and gigabit Ethernet fixed, mobile, tagged and authenticated ports. Link aggregate and tagged (trunked) link aggregate ports are not eligible for LPS monitoring and control.

1 Enable LPS on ports 6 through 12 on slot 3, 4, and 5 using the following command:

```
-> port-security 3/6-12 4/6-12 5/6-12 enable
```

2 Set the total number of learned MAC addresses allowed on the same ports to 25 using the following command:

```
-> port-security 3/6-12 4/6-12 5/6-12 maximum 25
```

3 Configure the amount of time in which source learning is allowed on all LPS ports to 30 minutes using the following command:

```
-> port-security shutdown 30
```

4 Select **shutdown** for the LPS violation mode using the following command:

```
-> port-security 3/6-12 4/6-12 5/6-12 violation shutdown
```

Note. *Optional*. To verify LPS port configurations, use the **show port-security** command. For example:

```
-> show port-security
  Port   Security    MaxMacs   Violation     IndividualMac     MacType
---------+---------+---------+-----------+------------------+-----------
   1/12   enabled    100       restrict   00:01:96:1c:f1:c0  dynamic
                                          00:06:5b:a3:19:3f  dynamic
   1/23   enabled      2       restrict   00:95:2a:0f:ce:19  configured
                                          00:95:2a:5e:cf:2a  configured
   1/24   enabled    100       shutdown

-> show port-security config-mac-range
   Port         LowMac           HighMac
---------+-------------------+-----------------
  1/12     00:00:00:00:00:00  ff:ff:ff:ff:ff:ff
  1/23     00:00:00:00:00:00  ff:ff:ff:ff:ff:ff
  1/24     00:95:2a:00:00:5a  00:95:2a:00:00:6f
```

To verify the source learning time limit value, use the **show port-security shutdown** command. For example:

```
-> show port-security shutdown
LPS Shutdown = 60 mins
```

Learned Port Security Overview

Learned Port Security (LPS) provides a mechanism for controlling network device access on one or more switch ports. Configurable LPS parameters allow the user to restrict the source learning of host MAC addresses to:

- A specific amount of time in which the switch allows source learning to occur on all LPS ports.
- A maximum number of learned MAC addresses allowed on the port.
- A list of configured authorized source MAC addresses allowed on the port.

Additional LPS functionality allows the user to specify how the LPS port handles unauthorized traffic. The following two options are available for this purpose:

- Block only traffic that violates LPS port restrictions; authorized traffic is forwarded on the port.
- Disable the LPS port when unauthorized traffic is received; all traffic is stopped and a port reset is required to return the port to normal operation.

LPS functionality is supported on the following 10/100 and Gigabit Ethernet port types:

- Fixed (non-mobile)
- Mobile
- 802.1Q tagged
- Authenticated

The following port types are not supported:

- 10 Gigabit Ethernet
- Link aggregate
- Tagged (trunked) link aggregate

How LPS Authorizes Source MAC Addresses

When a packet is received on a port that has LPS enabled, switch software checks the following criteria to determine if the source MAC address contained in the packet is allowed on the port:

- Is the source learning time window open?
- Is the number of MAC addresses learned on the port below the maximum number allowed?
- Is there a configured authorized MAC address entry for the LPS port that matches the packet's source MAC address?

Using the above criteria, the following table shows the conditions under which a MAC address is learned or blocked on an LPS port:

Time Limit	Max Number	Configured MAC	Result
Open	Below	No entry	No LPS violation; MAC learned
Closed	Below	No entry	LPS violation; MAC blocked
Open	Above	No entry	LPS violation; MAC blocked
Open	Below	Yes; entry matches	No LPS violation; MAC learned
Closed	Below	Yes; entry matches	No LPS violation; MAC learned
Open	Above	Yes; entry matches	LPS violation; MAC blocked
Open	Below	Yes; entry doesn't match	No LPS violation; MAC learned
Closed	Below	Yes; entry doesn't match	LPS violation; MAC blocked
Open	Above	Yes; entry doesn't match	LPS violation; MAC blocked

When a source MAC address violates any of the LPS conditions, the address is considered unauthorized. The LPS violation mode determines if the unauthorized MAC address is simply blocked on the port or if the entire port is disabled (see "Selecting the Security Violation Mode" on page 4-10). Regardless of which mode is selected, notice is sent to the Switch Logging task to indicate that a violation has occurred.

Dynamic Configuration of Authorized MAC Addresses

Once LPS authorizes the learning of a source MAC address, an entry containing the address and the port it was learned on is made in an LPS database table. This entry is then used as criteria for authorizing future traffic from this source MAC on that same port. In other words, learned authorized MAC addresses become configured criteria for an LPS port.

For example, if the source MAC address 00:da:95:00:59:0c is received on port 2/10 and meets the LPS restrictions defined for that port, then this address and its port are recorded in the LPS table. All traffic that is received on port 2/10 is compared to the 00:da:95:00:59:0c entry. If any traffic received on this port consists of packets that do not contain a matching source address, the packets are then subject to the LPS source learning time limit window and the maximum number of addresses allowed criteria.

When a dynamically configured MAC address is added to the LPS table, it does not become a configured MAC address entry in the LPS table until the switch configuration file is saved and the switch is rebooted. If a reboot occurs before this is done, all dynamically learned MAC addresses in the LPS table are cleared.

Static Configuration of Authorized MAC Addresses

It is also possible to statically configure authorized source MAC address entries into the LPS table. This type of entry behaves the same way as dynamically configured entries in that it authorizes port access to traffic that contains a matching source MAC address.

Static source MAC address entries, however, take precedence over dynamically learned entries. For example, if there are 2 static MAC address entries configured for port 2/1 and the maximum number allowed on port 2/1 is 10, then only 8 dynamically learned MAC addresses are allowed on this port.

Note that source learning of configured authorized MAC addresses is still allowed after the LPS time limit has expired. However, all learning is stopped if the number of MAC addresses learned meets or exceeds the maximum number of addresses allowed, even if the LPS time limit has not expired.

There are two ways to define a static source MAC address entry in the LPS table; specify an individual MAC address or a range of MAC addresses. See "Configuring Authorized MAC Addresses" on page 4-8 and "Configuring an Authorized MAC Address Range" on page 4-9 for more information.

Understanding the LPS Table

The LPS database table is separate from the source learning MAC address table. However, when a MAC is authorized for learning on an LPS port, an entry is made in the MAC address table in the same manner as if it was learned on a non-LPS port (see Chapter 2, "Managing Source Learning," for more information).

In addition to dynamic and configured source MAC address entries, the LPS table also provides the following information for each eligible LPS port:

- The LPS status for the port; enabled or disabled.
- The maximum number of MAC addresses allowed on the port.
- The violation mode selected for the port; restrict or shutdown.
- Statically configured MAC addresses and MAC address ranges.
- All MAC addresses learned on the port.
- The management status for the MAC address entry; configured or dynamic.

Note that dynamic MAC address entries become configured entries after the switch configuration is saved and the switch is rebooted. However, any dynamic MAC address entries that are not saved to the switch configuration are cleared if the switch reboots before the next save.

If the LPS port is shut down or the network device is disconnected from the port, the LPS table entries for this port are retained, but the source learning MAC address table entries for the same port are automatically cleared. In addition, if an LPS table entry is intentionally cleared from the table, the MAC address for this entry is automatically cleared from the source learning table at the same time.

To view the contents of the LPS table, use the **show port-security** command. Refer to the *OmniSwitch CLI Reference Guide* for more information about this command.

Enabling/Disabling Learned Port Security

By default, LPS is disabled on all switch ports. To enable LPS on a port, use the **port-security** command. For example, the following command enables LPS on port 1 of slot 4:

```
-> port-security 4/1 enable
```

To enable LPS on multiple ports, specify a range of ports or multiple slots. For example:

```
-> port-security 4/1-5 enable
-> port-security 5/12-20 6/10-15 enable
```

Note that when LPS is enabled on an active port, all MAC addresses learned on that port prior to the time LPS was enabled are cleared from the source learning MAC address table.

To disable LPS on a port, use the **port-security** command with the **disable** parameter. For example, the following command disables LPS on a range of ports:

```
-> port-security 5/21-24 6/1-4 disable
```

When LPS is disabled on a port, MAC address entries for that port are retained in the LPS table. The next time LPS is enabled on the port, the same LPS table entries are again active. If there is a switch reboot before the switch configuration is saved, however, dynamic MAC address entries are discarded from the table.

Use the **no** form of this command to disable LPS *and* clear all entries (configured and dynamic) in the LPS table for the specified port. For example:

```
-> no port-security 5/10
```

Configuring a Source Learning Time Limit

By default, the source learning time limit is disabled. Use the **port-security shutdown** command to set the number of minutes the source learning window is to remain open for LPS ports. While this window is open, source MAC addresses that comply with LPS port restrictions are authorized for learning on the related LPS port. The following actions trigger the start of the source learning timer:

- The **port-security shutdown** command. Each time this command is issued, the timer restarts even if a current window is still open or a previous window has expired.
- Switch reboot with a **port-security shutdown** command entry saved in the **boot.cfg** file.

The LPS source learning time limit is a switch-wide parameter that applies to all LPS enabled ports, not just one or a group of LPS ports. The following command example sets the time limit value to 30 minutes:

```
-> port-security shutdown time 30
```

Once the time limit value expires, source learning of any new dynamic MAC addresses is stopped on all LPS ports even if the number of addresses learned does not exceed the maximum allowed.

Note. Source learning of configured authorized MAC addresses is still allowed after the LPS time limit has expired; however, all learning is stopped if the number of MAC addresses learned meets or exceeds the maximum number of addresses allowed, even if the LPS time limit has not expired.

Configuring the Number of MAC Addresses Allowed

By default, one MAC address is allowed on an LPS port. To change this number, enter **port-security** followed by the port's *slot/port* designation then **maximum** followed by a number between 1 and 100. For example, the following command sets the maximum number of MAC addresses learned on port 10 of slot 6 to 75:

```
-> port-security 6/10 maximum 75
```

To specify a maximum number of MAC addresses allowed for multiple ports, specify a range of ports or multiple slots. For example:

```
-> port-security 1/10-15 maximum 10
-> port-security 2/1-5 4/2-8 5/10-14 maximum 25
```

Not that configured MAC addresses count towards the maximum number allowed. For example, if there are 10 configured authorized MAC addresses for an LPS port and the maximum number of addresses allowed is set to 15, then only 5 dynamically learned MAC address are allowed on this port.

If the maximum number of MAC addresses allowed is reached before the switch LPS time limit expires, then all source learning of dynamic *and* configured MAC addresses is stopped on the LPS port.

Configuring Authorized MAC Addresses

To configure a single source MAC address entry in the LPS table, enter **port-security** followed by the port's *slot/port* designation, then **mac** followed by a valid MAC address. For example, the following command configures a MAC address for port 4 on slot 6:

```
-> port-security 6/4 mac 00:20:da:9f:58:0c
```

To configure a single source MAC address entry for multiple ports, specify a range of ports or multiple slots. For example:

```
-> port-security 4/1-5 mac 00:20:95:41:2e:3f
-> port-security 5/12-20 6/10-15 mac 00:20:da:cf:59:4a
```

Use the **no** form of this command to clear configured *and/or* dynamic MAC address entries from the LPS table. For example, the following command removes a MAC address entry for port 12 of slot 4 from the LPS table:

```
-> port-security 4/12 no mac 00:20:95:00:fa:5c
```

Note that when a MAC address is cleared from the LPS table, it is automatically cleared from the source learning MAC address table at the same time.

Configuring an Authorized MAC Address Range

By default, each LPS port is set to a range of 00:00:00:00:00:00–ff:ff:ff:ff:ff:ff, which includes all MAC addresses. If this default is not changed, then addresses received on LPS ports are subject only to the source learning time limit and maximum number of MAC addresses allowed restrictions for the port.

To configure a source MAC address range for an LPS port, enter **port-security** followed by the port's *slot/port* designation, then **mac-range** followed by **low** and a MAC address, then **high** and a MAC address. For example, the following command configures a MAC address range for port 1 on slot 4:

```
-> port-security 4/1 mac low 00:20:da:00:00:10 high 00:20:da:00:00:50
```

To configure a source MAC address range for multiple ports, specify a range of ports or multiple slots. For example:

```
-> port-security 4/1-5 mac-range low 00:20:da:00:00:10 high 00:20:da:00:00:50
-> port-security 2/1-4 4/5-8 mac-range low 00:20:d0:59:0c:9a high
00:20:d0:59:0c:9f
```

To set the range back to the default values, enter **port-security** followed by the port's *slot/port* designation then **mac-range**. Leaving off the **low** and **high** MAC addresses will reset the range back to 00:00:00:00:00:00 and ff:ff:ff:ff:ff:ff. For example, the following command sets the authorized MAC address range to the default values for port 12 of slot 4:

```
-> port-security 4/12 mac-range
```

In addition, specifying a low end MAC and a high end MAC is optional. If either one is not specified, the default value is used. For example, the following commands set the authorized MAC address range on the specified ports to 00:da:25:59:0c:10–ff:ff:ff:ff:ff:ff and 00:00:00:00:00:00–00:da:25:00:00:9a:

```
-> port-security 2/8 mac-range low pp:da:25:59:0c
-> port-security 2/10 mac-range high 00:da:25:00:00:9a
```

Refer to the *OmniSwitch CLI Reference Guide* for more information about this command.

Selecting the Security Violation Mode

By default, the security violation mode for an LPS port is set to **restrict**. In this mode, when an unauthorized source MAC address is received on an LPS port, the packet containing the address is blocked. However, all other packets containing an authorized source MAC address are still allowed on the port.

Note that unauthorized source MAC addresses are not learned in the LPS table but are still recorded in the source learning MAC address table with a filtered operational status. This allows the user to view MAC addresses that were attempting unauthorized access to the LPS port.

The other violation mode option is **shutdown**. In this mode, the LPS port is disabled when an unauthorized MAC address is received; all traffic is prevented from forwarding on the port.

To configure the security violation mode for an LPS port, enter **port-security** followed by the port's *slot/port* designation, then **violation** followed by **restrict** or **shutdown**. For example, the following command selects the shutdown mode for port 1 on slot 4:

```
-> port-security 4/1 violation shutdown
```

To configure the security violation mode for multiple LPS ports, specify a range of ports or multiple slots. For example:

```
-> port-security 4/1-10 violation shutdown
-> port-security 1/10-15 2/1-10 violation restrict
```

Restoring the Operational State of an LPS Port

After a security violation occurs, the LPS port is either administratively disabled or is filtering traffic from one or more source MAC address. To return the port to normal operation without having to manually reset the port and/or module, use the **port-security release** command. For example:

```
-> port-security 4/1 release
-> port-security 1/10-15 2/1-10 release
```

When this command is used, all MAC addresses known to the specified port are flushed from the switch MAC address table.

Note. Using the **port-security release** command restores the port to the same operational state it was in prior to the security violation. This includes the activation of any existing LPS configuration for the port, LPS monitoring of the port is automatically restored.

Displaying Learned Port Security Information

To display LPS port and table information, use the show commands listed below:

show port-security Displays Learned Port Security configuration values as well as MAC addresses learned on the port.

show port-security shutdown Displays the current time limit value set for source learning on all LPS enabled ports.

For more information about the resulting display from these commands, see the *OmniSwitch CLI Reference Guide*. An example of the output for the **show port-security** and **show port-security shutdown** commands is also given in "Sample Learned Port Security Configuration" on page 4-3.

5 Configuring VLANs

In a flat bridged network, a broadcast domain is confined to a single LAN segment or even a specific physical location, such as a department or building floor. In a switch-based network, such as one comprised of Alcatel switching systems, a broadcast domain—or *VLAN*— can span multiple physical switches and can include ports from a variety of media types. For example, a single VLAN could span three different switches located in different buildings and include Ethernet, Gigabit, 802.1q tagged ports or a link aggregate of ports.

In This Chapter

This chapter describes how to define and manage VLAN configurations through the Command Line Interface (CLI). CLI commands are used in the configuration examples; for more details about the syntax of commands, see the *OmniSwitch CLI Reference Guide*.

Configuration procedures described in this chapter include:

- "Creating/Modifying VLANs" on page 5-6.
- "Defining VLAN Port Assignments" on page 5-7.
- "Enabling/Disabling Spanning Tree for a VLAN" on page 5-10.
- "Enabling/Disabling VLAN Authentication" on page 5-11.
- "Configuring VLAN Router Interfaces" on page 5-11.
- "Bridging VLANs Across Multiple Switches" on page 5-14.
- "Verifying the VLAN Configuration" on page 5-16

For information about statically and dynamically assigning switch ports to VLANs, see Chapter 7, "Assigning Ports to VLANs."

For information about defining VLAN rules that allow dynamic assignment of mobile ports to a VLAN, see Chapter 8, "Defining VLAN Rules."

For information about Spanning Tree, see Chapter 6, "Configuring Spanning Tree Parameters."

For information about routing, see Chapter 13, "Configuring IP." and Chapter 19, "Configuring IPX."

For information about Layer 2 VLAN authentication, see Chapter 21, "Configuring Authenticated VLANs."

VLAN Specifications

RFCs Supported	2674 - *Definitions of Managed Objects for Bridges with Traffic Classes, Multicast Filtering and Virtual LAN Extensions*
IEEE Standards Supported	802.1Q - *Virtual Bridged Local Area Networks* 802.1D - *Media Access Control Bridges*
Maximum VLANs per switch	4094 (including default VLAN 1)
Maximum VLAN port associations	32768
Maximum router VLANs per switch	4094 IP and 256 IPX (single MAC router mode) 64 IP, IPX or combination of both (multiple MAC router mode)
Maximum IP router interfaces per VLAN	8
Maximum IP router interfaces per switch	4096
Maximum Spanning Tree VLANs per switch or stack	253
Maximum authenticated VLANs per switch	128
CLI Command Prefix Recognition	All VLAN management commands support prefix recognition. See the "Using the CLI" chapter in the *OmniSwitch 7700/7800/8800 Switch Management Guide* for more information.

VLAN Defaults

Parameter Description	Command	Default
VLAN identifier (VLAN ID)	**vlan**	VLAN 1 predefined on each switch.
VLAN administrative state	**vlan**	Enabled
VLAN description	**vlan name**	VLAN identifier (VLAN ID)
VLAN Spanning Tree state	**vlan stp**	Enabled
VLAN IP router interface	**ip interface**	VLAN 1 router interface.
VLAN IPX router interface	**vlan router ipx**	No router interface defined.
Switch MAC router mode	**vlan router mac multiple**	Multiple MAC router mode
VLAN authentication status	**vlan authentication**	Disabled
VLAN port associations	**vlan port default**	All ports initially associated with default VLAN 1.

Sample VLAN Configuration

The following steps provide a quick tutorial that will create VLAN 255 with a user-defined description, IP router interface, and static switch port assignments:

Note. *Optional.* Creating a new VLAN involves specifying a VLAN ID that is not already assigned to an existing VLAN. To determine if a VLAN already exists in the switch configuration, enter **show vlan**. If VLAN 255 does not appear in the **show vlan** output, then it does not exist on the switch. For example,

```
-> show vlan
                          stree                       mble
 vlan     admin   oper   1x1  flat   auth   ip   ipx   tag   name
+------+--------+-------+----------+------+----+-----+-----+----------+
    1       on      off   on    on   off   off    off  off   VLAN 1
    2       on      off   on   off   off   off    off  off   VLAN 2
    3       on      off   off  off   off   off    off  off   VLAN 3
    4       on      off   off   on   off   off    off  off   VLAN 4
    5       on      off   on    on   off   off    off  off   VLAN 5
```

1 Create VLAN 255 with a description of Finance IP Network using the **vlan** command. For example:

```
-> vlan 255 name "Finance IP Network"
```

2 Define a IP router interface, named Finance, using the **ip interface** command to assign an IP host address to VLAN 255 that will enable routing of IP traffic to other IP router VLANs. For example:

```
-> ip interface Finance address 21.0.0.10 vlan 255
```

3 Define an IPX router interface using the **vlan router ipx** command to assign an IPX address to VLAN 255 that will enable routing of IPX traffic to other IPX router VLANs. For example:

```
-> vlan 255 router ipx 2a10
```

4 Assign switch ports 2 through 5 on slot 3 to VLAN 255 using the following command:

```
-> vlan 255 port default 3/2-5
```

Note. *Optional*. To verify the VLAN 255 configuration, enter **show vlan** followed by 255. For example:

```
-> show vlan 255
    Name                : Finance IP Network,
    Administrative State: enabled
    Operational State   : enabled
    1x1 Spanning Tree State  : enabled,
    Flat Spanning Tree State : enabled,
    Authentication      : disabled,
    IP Router Port      : on,
    IPX Router Port     : 2a10 active e2 ticks:0,
    Mobile Tag          : off
```

To verify that ports 2 through 5 on slot 3 were assigned to VLAN 255, enter **show vlan** followed by 255 then **port**. For example:

```
-> show vlan 255 port
  port     type      status
--------+---------+--------------
   3/2    default   inactive
   3/3    default   inactive
   3/4    default   inactive
   3/5    default   inactive
```

VLAN Management Overview

One of the main benefits of using VLANs to segment network traffic, is that VLAN configuration and port assignment is handled through switch software. This eliminates the need to physically change a network device connection or location when adding or removing devices from the VLAN broadcast domain. The VLAN management software handles the following VLAN configuration tasks and reporting performed on an Alcatel switch:

- Creating or modifying VLANs.
- Assigning or changing default VLAN port associations (VPAs).
- Enabling or disabling VLAN participation in the current Spanning Tree algorithm.
- Enabling or disabling VLAN authentication.
- Defining VLAN IPX router interfaces to enable routing of VLAN IPX traffic.
- Enabling or disabling unique MAC address assignments for each router VLAN defined.
- Displaying VLAN configuration information.

In addition to the above tasks, VLAN management software tracks and reports the following information to other switch software applications:

- VLAN configuration changes, such as adding or deleting VLANs, modifying the status of VLAN properties (e.g., administrative, Spanning Tree, and authentication status), changing the VLAN description, or configuring VLAN router interfaces.
- VLAN port association changes triggered by VLAN management and other switch software applications, such as 802.1Q VLAN tagging and dynamic mobile port assignment.
- The VLAN operational state, which does not become active until at least one active switch port is associated with the VLAN.

Creating/Modifying VLANs

The initial configuration for all Alcatel switches consists of a default VLAN 1 and all switch ports are initially assigned to this VLAN. When a switching module is added to the switch, the module's physical ports are also assigned to VLAN 1. If additional VLANs are not configured on the switch, then the entire switch is treated as one large broadcast domain. All ports will receive all traffic from all other ports.

Alcatel switches support up to 4094 VLANs on one switch, including default VLAN 1. In compliance with the IEEE 802.1Q standard, each VLAN is identified by a unique number, referred to as the *VLAN ID*. The user specifies a VLAN ID to create, modify or remove a VLAN and to assign switch ports to a VLAN. When a packet is received on a port, the port's VLAN ID is inserted into the packet. The packet is then bridged to other ports that are assigned to the same VLAN ID. In essence, the VLAN broadcast domain is defined by a collection of ports and packets assigned to its VLAN ID.

The operational status of a VLAN remains inactive until at least one active switch port is assigned to the VLAN. This means that VLAN properties, such as Spanning Tree or router interfaces, also remain inactive. Ports are considered active if they are connected to an active network device. Non-active port assignments are allowed, but do not change the VLAN's operational state.

Ports are either statically or dynamically assigned to VLANs. When a port is assigned to a VLAN, a VLAN port association (VPA) is created and tracked by VLAN management switch software. For more information about VPAs, see "Defining VLAN Port Assignments" on page 5-7 and Chapter 7, "Assigning Ports to VLANs."

Adding/Removing a VLAN

To add a VLAN to the switch configuration, enter **vlan** followed by a unique VLAN ID number between 2 and 4094, an optional administrative status, and an optional description. For example, the following command creates VLAN 755 with a description:

```
-> vlan 755 enable name "IP Finance Network"
```

By default, administrative status and Spanning Tree are enabled when the VLAN is created and the VLAN ID is used for the description if one is not specified. Note that quotation marks are required if the description contains multiple words separated by spaces. If the description consists of only one word or multiple words separated by another character, such as a hyphen, then quotes are not required.

To remove a VLAN from the switch configuration, use the **no** form of the **vlan** command.

```
-> no vlan 755
```

When a VLAN is deleted, any router interfaces defined for the VLAN are removed and all VLAN port associations are dropped. For more information about router interfaces, see "Configuring VLAN Router Interfaces" on page 5-11.

To view a list of VLANs already configured on the switch, use the **show vlan** command. See "Verifying the VLAN Configuration" on page 5-16 for more information.

Enabling/Disabling the VLAN Administrative Status

To enable or disable the administrative status for an existing VLAN, enter **vlan** followed by an existing VLAN ID and either **enable** or **disable**.

```
-> vlan 755 disable
-> vlan 255 enable
```

When the administrative status for a VLAN is disabled, VLAN port assignments are retained but traffic is not forwarded on these ports. If any rules were defined for the VLAN, they are also retained and continue to classify mobile port traffic. For more information about VLAN rules, see Chapter 8, "Defining VLAN Rules."

Modifying the VLAN Description

To change the description for a VLAN, enter **vlan** followed by an existing VLAN ID and the keyword **name** followed by the new description (up to 32 characters). For example, the following command changes the description for VLAN 455 to "Marketing IP Network":

```
-> vlan 455 name "Marketing IP Network"
```

Note that quotation marks are required if the description consists of multiple words separated by spaces. If the description consists of only one word or words are separated by another character, such as a hyphen, then quotes are not required. For example,

```
-> vlan 455 name Marketing-IP-Network
```

Defining VLAN Port Assignments

Alcatel switches support static and dynamic assignment of physical switch ports to a VLAN. Regardless of how a port is assigned to a VLAN, once the assignment occurs, a VLAN port association (VPA) is created and tracked by VLAN management software on each switch. To view current VLAN port assignments in the switch configuration, use the **show vlan port** command.

Methods for statically assigning ports to VLANs include the following:

- Using the **vlan port default** command to define a new configured default VLAN for both non-mobile (fixed) and mobile ports. (See "Changing the Default VLAN Assignment for a Port" on page 5-8.)

- Using the **vlan 802.1q** command to define tagged VLANs for non-mobile ports. This method allows the switch to bridge traffic for multiple VLANs over one physical port connection. (See Chapter 10, "Configuring 802.1Q.")

- Configuring ports as members of a link aggregate that is assigned to a configured default VLAN. (See Chapter 11, "Configuring Static Link Aggregation," and Chapter 12, "Configuring Dynamic Link Aggregation.")

Dynamic assignment applies only to mobile ports. When traffic is received on a mobile port, the packets are examined to determine if their content matches any VLAN rules configured on the switch. If a match occurs, the mobile port is automatically assigned to the VLAN without user intervention.

For more information about mobile ports and traffic rules, see Chapter 7, "Assigning Ports to VLANs," and Chapter 8, "Defining VLAN Rules."

Changing the Default VLAN Assignment for a Port

To assign a switch port to a new default VLAN, enter **vlan** followed by an existing VLAN ID number, **port default**, then the slot/port designation. For example, the following command assigns port 5 on slot 2 to VLAN 955:

```
-> vlan 955 port default 2/5
```

All ports initially belong to default VLAN 1. When the **vlan port default** command is used, the port's default VLAN assignment is changed to the specified VLAN. In the above example, VLAN 955 is now the default VLAN for port 5 on slot 2 and this port is no longer associated with VLAN 1.

The **vlan port default** command is also used to change the default VLAN assignment for an aggregate of ports. The link aggregate control number is specified instead of a slot and port. For example, the following command assigns link aggregate 10 to VLAN 755:

```
-> vlan 755 port default 10
```

For more information about configuring an aggregate of ports, see Chapter 11, "Configuring Static Link Aggregation," and Chapter 12, "Configuring Dynamic Link Aggregation."

Use the **no** form of the **vlan port default** command to remove a default VPA. When this is done, VLAN 1 is restored as the port's default VLAN.

```
-> vlan 955 no port default 2/5
```

Configuring Dynamic VLAN Port Assignment

Configuring the switch to allow dynamic VLAN port assignment requires the following steps:

1 Use the **vlan port mobile** command to enable mobility on switch ports that will participate in dynamic VLAN assignment. See Chapter 7, "Assigning Ports to VLANs,"for detailed procedures.

2 Enable/disable mobile port properties that determine mobile port behavior. See Chapter 7, "Assigning Ports to VLANs," for detailed procedures.

3 Create VLANs that will receive and forward mobile port traffic. See "Adding/Removing a VLAN" on page 5-6 for more information.

4 Define VLAN rules for the VLANs that were created in Step 3. These rules will trigger dynamic assignment of mobile ports to these VLANs. See "Configuring VLAN Rule Classification" on page 5-9 .

Once the above configuration steps are completed, dynamic VLAN assignment occurs when a device connected to a mobile port starts to send traffic. This traffic is examined by switch software to determine which VLAN should carry the traffic based on VLAN rule criteria.

See Chapter 7, "Assigning Ports to VLANs," and Chapter 8, "Defining VLAN Rules," for more information and examples of dynamic VLAN port assignment.

Configuring VLAN Rule Classification

VLAN rule classification triggers dynamic VLAN port assignment when traffic received on a mobile port matches the criteria defined in a VLAN rule. Different rule types are available for classifying different types of network device traffic. It is possible to define multiple rules for one VLAN and rules for multiple VLANs. However, only IP and IPX protocol rules support the dynamic assignment of one mobile port to multiple VLANs.

The following table provides a list of commands used to define the various types of VLAN rules. For more detailed information about rule criteria and classification, see Chapter 8, “Defining VLAN Rules.”

Rule Types	Command
DHCP	**vlan dhcp mac** **vlan dhcp mac range** **vlan dhcp port** **vlan dhcp generic**
Binding	**vlan binding mac-ip-port** **vlan binding mac-port-protocol** **vlan binding mac-port** **vlan binding mac-ip** **vlan binding ip-port** **vlan binding port-protocol**
MAC address	**vlan mac** **vlan mac range**
Network address	**vlan ip** **vlan ipx**
Protocol	**vlan protocol**
Custom (user-defined)	**vlan user**
Port	**vlan port**

Enabling/Disabling Spanning Tree for a VLAN

When a VLAN is created, an 802.1D standard Spanning Tree Algorithm and Protocol (STP) instance is enabled for the VLAN by default. The spanning tree operating mode set for the switch determines how VLAN ports are evaluated to identify redundant data paths.

If the Spanning Tree switch operating mode is set to *flat*, then VLAN port connections are checked against other VLAN port connections for redundant data paths. In this mode, if the Spanning Tree is disabled on VLAN 1, then it is disabled for all configured VLANs. However, disabling Spanning Tree on a VLAN other than VLAN 1 excludes only those ports associated with that VLAN from Spanning Tree control.

If the Spanning Tree switch operating mode is set to *1x1*, there is a single Spanning Tree instance for each VLAN broadcast domain. Disabling Spanning Tree on a VLAN in this mode excludes ports associated with that VLAN from Spanning Tree control.

Note. When Spanning Tree is disabled for a VLAN, all active ports associated with that VLAN are transitioned to a forwarding state. Ensure that disabling Spanning Tree fora particular VLAN will not cause a network loop to go undetected.

The **vlan stp** command is used to enable/disable a Spanning Tree instance for an existing VLAN. In the following examples, Spanning Tree is disabled on VLAN 255 and enabled on VLAN 755:

```
-> vlan 255 stp disable
-> vlan 755 stp enable
```

The above commands configure the VLAN Spanning Tree status for both the 1x1 and flat Spanning Tree modes. Using the **1x1** or **flat** parameter with this command, configures the STP status only for the mode specified by the parameter. For example, the following command configures a disabled Spanning Tree status for VLAN 755 that applies only when the switch is operating in the flat Spanning Tree mode:

```
-> vlan 755 flat stp disable
```

As a result of the above command, Spanning Tree is active on VLAN 755 when the switch is operating in the 1x1 mode, but inactive on VLAN 755 when the switch is operating in the flat mode.

Note that up to 253 Spanning Tree instances per switch are supported. Therefore, when the switch is operating in the 1x1 mode, only 253 VLANs can have an active Spanning Tree instance at any given time.

STP does not become operationally active on a VLAN unless the VLAN is operationally active, which occurs when at least one active port is assigned to the VLAN. Also, STP is enabled/disabled on individual ports. So even if STP is enabled for the VLAN, a port assigned to that VLAN must also have STP enabled. See Chapter 6, “Configuring Spanning Tree Parameters,” for more information on the Alcatel Spanning Tree implementation.

Enabling/Disabling VLAN Authentication

Layer 2 authentication uses VLAN membership to grant access to network resources. Authenticated VLANs control membership through a log-in process; this is sometimes called *user authentication*. A VLAN must have authentication enabled before it can participate in the Layer 2 authentication process.

To enable/disable authentication on an existing VLAN, use the **vlan authentication** command. For example, the following commands enable authentication on VLAN 955 and disable it on VLAN 455:

```
-> vlan 955 authentication enable
-> vlan 455 authentication disable
```

Once authentication is enabled on a VLAN, then only authenticated mobile port devices can join the VLAN after completing the appropriate log-in process. To enable authentication on a mobile port, use the **vlan port mobile** command. For more information about mobile port commands and Layer 2 authentication for Alcatel switches, see Chapter 7, "Assigning Ports to VLANs," and Chapter 21, "Configuring Authenticated VLANs."

Configuring VLAN Router Interfaces

Network device traffic is bridged (switched) at the Layer 2 level between ports that are assigned to the same VLAN. However, if a device needs to communicate with another device that belongs to a different VLAN, then Layer 3 routing is necessary to transmit traffic between the VLANs. Bridging makes the decision on where to forward packets based on the packet's destination MAC address; routing makes the decision on where to forward packets based on the packet's IP or IPX network address (e.g., IP - 21.0.0.10, IPX - 210A).

Alcatel switches support routing of IP and IPX traffic. A VLAN is available for routing when at least one router interface is defined for that VLAN and at least one active port is associated with the VLAN. Up to eight IP interfaces and one IPX interface can be configured for each VLAN. The maximum number of IP interfaces allowed for the entire switch is 4096.

If a VLAN does not have a router interface, the ports associated with that VLAN are in essence firewalled from other VLANs. For information about how to configure router interfaces, see Chapter 13, "Configuring IP," and "Configuring an IPX Router Interface" on page 5-12.

The number of VLANs that can have routing configured depends on which MAC router mode is active on the switch (multiple or single mode). If the switch is running in multiple MAC router mode (the default), then a maximum of 64 VLANs can have IP, IPX, or a combination of both router interfaces defined. If the switch is running in single MAC router mode, a maximum of 4094 VLANs can have IP interfaces defined and a maximum of 256 VLANs can have IPX interfaces defined.

Use the **show vlan router mac status** command to determine which MAC router mode is active on the switch. See "Enabling/Disabling Multiple MAC Router Mode" on page 5-13 for more information about MAC router modes.

Configuring an IPX Router Interface

Use the **vlan router ipx** command to define an IPX router interface for an existing VLAN. Specify the following when using this command:

1 The VLAN ID of the router VLAN (can only specify an existing VLAN).

2 The IPX network address to assign to the router interface. An IPX network address consists of eight hex characters (e.g., 4001690D or 0000210A). If less than eight hex digits are specified, the address is prefixed with zeros to equal eight digits. For example, if **950A** is entered, the actual IPX network address value is **0000950A**.

3 Select one of the following keywords to change the advertisement mode. By default, the advertisement mode is set to active (RIP and SAP updates are processed):

IPX advertisement mode keywords	
rip	**inactive**
active	**triggered**

4 IPX router encapsulation (defaults to Ethernet-II). Select one of the following keywords to change the encapsulation:

IPX encapsulation keywords	
e2 or ethernet2	**llc**
novell	**snap**

5 A 16-bit value between 0 (the default) and 65535 that specifies the number of ticks for the IPX delay time. A tick is approximately 1/18th of a second.

The following **vlan router ipx** command example configures an IPX router interface for VLAN 955 with an IPX network address of 0000950A that will process RIP and SAP updates, use Ethernet-II encapsulation when generating packets, and have a zero tick delay time value:

```
-> vlan 955 router ipx 950A active e2 timeticks 0
```

Specifying the advertisement mode, encapsulation, and delay time value in ticks is optional, so it is not necessary to enter these parameters as part of the command to accept their default values. For example, either one of the following commands will create an IPX router interface for VLAN 855 with the same properties:

```
-> vlan 855 router ipx 8500100A active e2 timeticks 0
-> vlan 855 router ipx 8500100A
```

To remove an IPX router interface from a VLAN, use the **no** form of the **vlan router ipx** command.

```
-> vlan 855 no router ipx
```

Modifying an IPX Router Interface

The **vlan router ipx** command is also used to modify one or more existing IPX router interface parameter values. For example, the following command changes the existing router interface IPX address for VLAN 955 to 1000450C:

```
-> vlan 955 router ipx 1000450C
```

It is not necessary to first remove the IPX router interface from the VLAN. The changes specified will overwrite existing parameter values. For example, the following command changes the advertisement mode to RIP only, the encapsulation to LLC, and the delay time value to 1500. The IPX address is not changed in this example, but is required as part of the command syntax to identify a change to the router interface:

```
-> vlan 955 router ipx 1000450C rip llc timeticks 10
```

Use the **show vlan** command to verify IPX router changes. For more information about this command, see the *OmniSwitch CLI Reference Guide.*

Enabling/Disabling Multiple MAC Router Mode

Multiple MAC router mode is enabled by default and is required for compatibility when interoperating with legacy OmniSwitch/Router (OS/R) products. When a switch is operating in this mode, a unique MAC address is assigned to each router VLAN. This limits the number of router VLANs per switch to 64 IP, IPX, or a combination of both (IP and IPX router interfaces on the same VLAN share the same MAC address).

Use the **vlan router mac multiple** command to enable or disable multiple MAC router mode for the switch. For example:

```
-> vlan router mac multiple disable
```

Note. Enabling or disabling multiple MAC router mode is supported only when the switch is rebooted. To configure a mode change to occur at boot time, add the **vlan router mac multiple enable** or **vlan router mac multiple disable** command to the **boot.cfg** file. For more information about editing the **boot.cfg** file, see the “Working With Configuration Files” chapter in the *OmniSwitch 7700/7800/8800 Switch Management Guide*.

When determining the appropriate MAC router mode to use, consider the following:

- Each switch comes equipped with 32 MAC addresses that are allocated for various switch resources, such as router VLANs. To define more than 32 router VLANs on a switch operating in the multiple MAC router mode, additional MAC addresses are required. Contact your Alcatel representative for information about how to increase the number of available MAC addresses.
- If multiple MAC router mode is disabled, then the switch runs in single MAC router mode. In this mode, each router VLAN is assigned the same MAC address, which is the base chassis MAC address for the switch. This increases the number of router VLANs supported on each switch to 4094 IP and 256 IPX. In this case, it is not necessary to allocate additional switch MAC addresses to define more than 32 router VLANs.
- Regardless of which mode the switch is running in, VLAN 1 router interfaces are always assigned the base chassis MAC address.
- Enabling multiple MAC router mode on a switch that has more than 64 router VLANs configured is not allowed. If an attempt is made to do so, an error message is displayed and the switch remains operating in the single MAC router mode.

To determine which mode the switch is currently operating in, the total number of VLANs configured on the switch, and the number of router VLANs configured, use the **show vlan router mac status** command. For more information about this command, see the *OmniSwitch CLI Reference Guide*.

Bridging VLANs Across Multiple Switches

To create a VLAN *bridging domain* that extends across multiple switches:

1 Create a VLAN on each switch with the same VLAN ID number (e.g., VLAN 10).

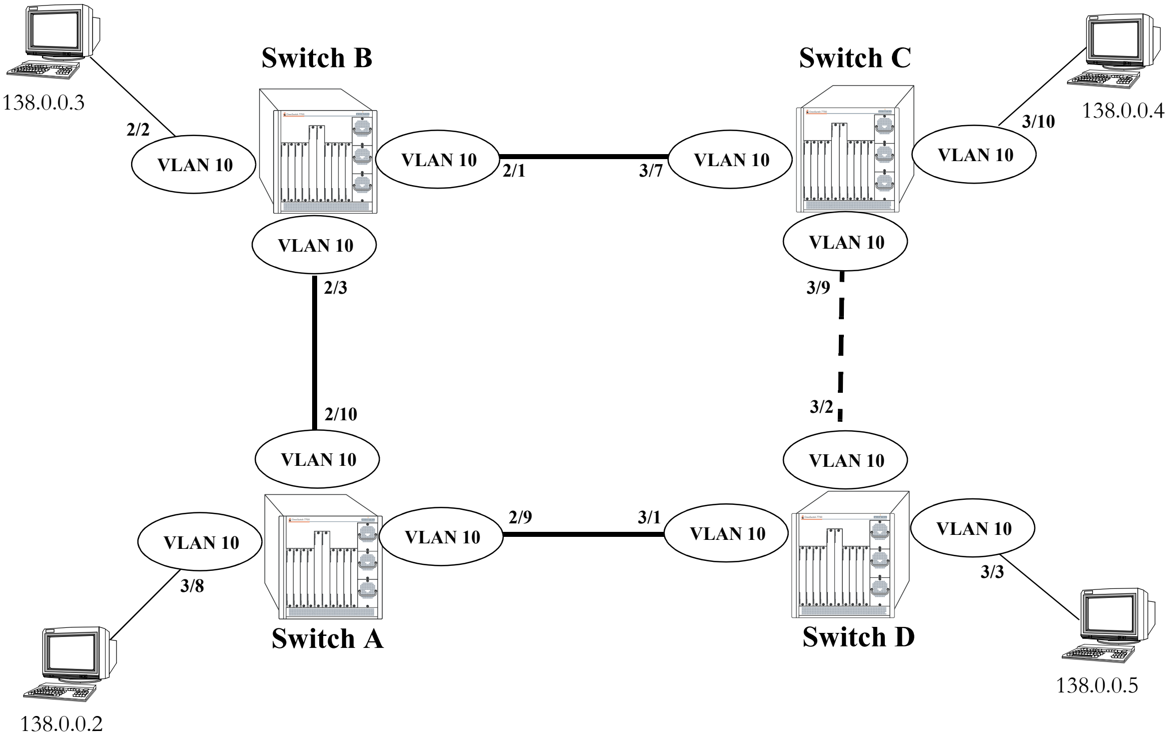
2 If using mobile ports for end user device connections, define VLAN rules that will classify mobile port traffic into the VLAN created in Step 1.

3 On each switch, assign the ports that will provide connections to other switches to the VLAN created in Step 1.

4 On each switch, assign the ports that will provide connections to end user devices (e.g., workstations) to the VLAN created in Step 1. (If using mobile ports, this step will occur automatically when the device connected to the mobile port starts to send traffic.)

5 Connect switches and end user devices to the assigned ports.

The following diagram shows the physical configuration of an example VLAN bridging domain:

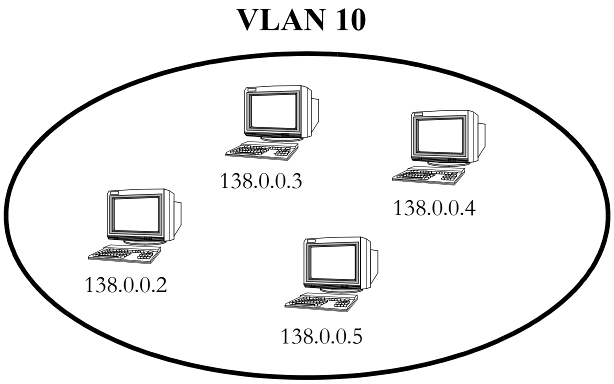


VLAN Bridging Domain: Physical Configuration

In the above diagram, VLAN 10 exists on all four switches and the connection ports between these switches are assigned to VLAN 10. The workstations can communicate with each other because the ports to which they are connected are also assigned to VLAN 10. It is important to note that connection cables do not have to connect to the same slot/port on each switch. The key is that the slot/port must belong to the same VLAN on each switch. To carry multiple VLANs between switches across a single physical connection cable, use the 802.1Q tagging feature (see Chapter 10, “Configuring 802.1Q”).

The connection between Switch C and D is shown with a broken line because the ports that provide this connection are in a blocking state. Spanning Tree is active by default on all switches, VLANs and ports. The Spanning Tree algorithm determined that if all connections between switches were active, a network loop would exist that could cause unnecessary broadcast traffic on the network. The path between Switch C and D was shut down to avoid such a loop. See Chapter 6, “Configuring Spanning Tree Parameters,” for information about how Spanning Tree configures network topologies that are loop free.

The following diagram shows the same bridging domain example as seen by the end user workstations. Because traffic between these workstations is *transparently bridged* across physical switch connections within the VLAN 10 domain, the workstations are basically unaware that the switches even exist. Each workstation believes that the others are all part of the same VLAN, even though they are physically connected to different switches.



VLAN Bridging Domain: Logical View

Creating a VLAN bridging domain across multiple switches allows VLAN members to communicate with each other, even if they are not connected to the same physical switch. This is how a logical grouping of users can traverse a physical network setup without routing and is one of the main benefits of using VLANs.

Verifying the VLAN Configuration

To display information about the VLAN configuration on the switch, use the show commands listed below:

show vlan Displays a list of all VLANs configured on the switch and the status of related VLAN properties (e.g., admin, Spanning Tree, and router interface status).

show vlan port Displays a list of VLAN port assignments.

show ip interface Displays the IP router interface configuration.

show vlan router mac status Displays the current MAC router operating mode (single or multiple) and router VLAN statistics.

For more information about the resulting displays from these commands, see the *OmniSwitch CLI Reference Guide*. An example of the output for the **show vlan** and **show vlan port** commands is also given in "Sample VLAN Configuration" on page 5-3.

6 Configuring Spanning Tree Parameters

The Spanning Tree Algorithm and Protocol (STP) is a self-configuring algorithm that maintains a loop-free topology while providing data path redundancy and network scalability. Based on the IEEE 802.1D standard, the Alcatel STP implementation distributes the Spanning Tree load between the Chassis Management Module (CMM) and the Network Interface modules (NIs). This ensures a Spanning Tree that continues to respond to STP Bridge Protocol Data Units (BPDU) received on switch ports and port link up and down states in the event of a CMM fail over to a backup CMM. In addition, the Alcatel distributed implementation incorporates the following Spanning Tree features:

- Configures a physical topology into a single Spanning Tree to ensure that there is only one data path between any two switches.
- Supports fault tolerance within the network topology. The Spanning Tree is reconfigured in the event of a data path or bridge failure or when a new switch is added to the topology.
- Supports two Spanning Tree operating modes; *flat* (single STP instance per switch) and *1x1* (single STP instance per VLAN).
- Supports three Spanning Tree Algorithms; 802.1D (STP), 802.1w (RSTP), and 802.1s (MSTP).
- Allows 802.1Q tagged ports and link aggregate logical ports to participate in the calculation of the STP topology.

The Distributed Spanning Tree software is active on all switches by default. As a result, a loop-free network topology is automatically calculated based on default Spanning Tree switch, bridge, and port parameter values. It is only necessary to configure Spanning Tree parameters to change how the topology is calculated and maintained.

In This Chapter

This chapter provides an overview about how Spanning Tree works and how to configure Spanning Tree parameters through the Command Line Interface (CLI). CLI commands are used in the configuration examples; for more details about the syntax of commands, see the *OmniSwitch CLI Reference Guide*.

Configuration procedures described in this chapter include:

- Selecting the switch Spanning Tree operating mode (flat or 1x1) on page 6-10.
- Configuring Spanning Tree bridge parameters on page 6-13.
- Configuring Spanning Tree port parameters on page 6-21.
- Configuring an example Spanning Tree topology on page 6-30

Spanning Tree Specifications

IEEE Standards supported	802.1D–*Media Access Control (MAC) Bridges* 802.1w–*Rapid Reconfiguration (802.1D Amendment 2)* 802.1Q–*Virtual Bridged Local Area Networks* 802.1s–*Multiple Spanning Trees (802.1Q Amendment 3)*
Spanning Tree Operating Modes supported	Flat mode - one spanning tree instance per switch 1x1 mode - one spanning tree instance per VLAN
Spanning Tree Protocols supported	802.1D Standard Spanning Tree Algorithm and Protocol (STP) 802.1w Rapid Spanning Tree Algorithm and Protocol (RSTP) 802.1s Multiple Spanning Tree Protocol (MSTP)
Spanning Tree port eligibility	Fixed ports (non-mobile) 802.1Q tagged ports Link aggregate of ports
Maximum 1x1 mode Spanning Tree instances per switch	253
Maximum flat mode 802.1s Multiple Spanning Tree Instances (MSTI) per switch	16 MSTI, in addition to the Common and Internal Spanning Tree instance (also referred to as MSTI 0).
CLI Command Prefix Recognition	All Spanning Tree commands support prefix recognition. See the "Using the CLI" chapter in the *OmniSwitch 7700/7800/8800 Switch Management Guide* for more information.

Spanning Tree Bridge Parameter Defaults

Parameter Description	Command	Default
Spanning Tree operating mode	**bridge mode**	1x1 (a separate Spanning Tree instance for each VLAN)
Spanning Tree protocol	**bridge protocol**	STP (802.1D)
BPDU switching status.	**bridge bpdu-switching**	Disabled
Priority value for the Spanning Tree instance.	**bridge priority**	32768
Hello time interval between each BPDU transmission.	**bridge hello time**	2 seconds
Maximum aging time allowed for Spanning Tree information learned from the network.	**bridge max age**	20 seconds
Spanning Tree port state transition time.	**bridge forward delay**	15 seconds

Spanning Tree Port Parameter Defaults

Parameter Description	Command	Default
Spanning Tree port administrative state	**bridge slot/port**	Enabled
Spanning Tree port priority value	**bridge slot/port priority**	7
Spanning Tree port path cost.	**bridge slot/port path cost**	0 (cost is based on port speed)
Path cost mode	**bridge path cost mode**	Auto (16-bit in 1x1 mode and 802.1D or 802.1w flat mode, 32-bit in 802.1s flat mode)
Port state management mode	**bridge slot/port mode**	Dynamic (Spanning Tree Algorithm determines port state)
Type of port connection	**bridge slot/port connection**	auto point to point

Multiple Spanning Tree (MST) Region Defaults

Although the following parameter values are specific to the MSTP (802.1s), they are configurable regardless of which mode (flat or 1x1) or protocol is active on the switch.

Parameter Description	Command	Default
The MST region name	**bridge mst region name**	blank
The revision level for the MST region	**bridge mst region revision level**	0
The maximum number of hops authorized for the region	**bridge mst region max hops**	20
The number of Multiple Spanning Tree Instances (MSTI).	**bridge msti**	1 (flat mode instance)
The VLAN to MSTI mapping.	**bridge msti vlan**	All VLANs are mapped to the Common Internal Spanning Tree (CIST) instance

Spanning Tree Overview

Alcatel switches support the use of the 802.1D Spanning Tree Algorithm and Protocol (STP), the 802.1w Rapid Spanning Tree Algorithm and Protocol (RSTP), and the 802.1s Multiple Spanning Tree Protocol (MSTP).

RSTP expedites topology changes by allowing blocked ports to transition directly into a forwarding state, bypassing listening and learning states. This provides rapid reconfiguration of the Spanning Tree in the event of a network path or device failure.

The 802.1w standard is an amendment to the 802.1D document, thus RSTP is based on STP. Regardless of which one of these two protocols a switch or VLAN is running, it can successfully interoperate with other switches or VLANs.

MSTP is an enhancement to the 802.1Q Common Spanning Tree (CST), which is provided when an Alcatel switch is running in the flat Spanning Tree operating mode. The flat mode applies a single spanning tree instance across all VLAN port connections on a switch. MSTP allows the configuration of Multiple Spanning Tree Instances (MSTIs) in addition to the CST instance. Each MSTI is mapped to a set of VLANs. As a result, flat mode can now support the forwarding of VLAN traffic over separate data paths.

This section provides a Spanning Tree overview based on RSTP operation and terminology. Although MSTP is based on RSTP, see Chapter 3, “Using 802.1s Multiple Spanning Tree,” for specific information about configuring MSTP.

How the Spanning Tree Topology is Calculated

The *tree* consists of links and bridges that provide a single data path that spans the bridged network. At the base of the tree is a *root bridge*. One bridge is elected by all the bridges participating in the network to serve as the root of the tree. After the root bridge is identified, STP calculates the best path that leads from each bridge back to the root and blocks any connections that would cause a network loop.

To determine the best path to the root, STP uses the *path cost* value, which is associated with every port on each bridge in the network. This value is a configurable weighted measure that indicates the contribution of the port connection to the entire path leading from the bridge to the root.

In addition, a *root path cost* value is associated with every bridge. This value is the sum of the path costs for the port that receives frames on the best path to the root (this value is zero for the root bridge). The bridge with the lowest root path cost becomes the *designated bridge* for the LAN, as it provides the shortest path to the root for all bridges connected to the LAN.

During the process of calculating the Spanning Tree topology, each port on every bridge is assigned a *port role* based on how the port and/or its bridge will participate in the active Spanning Tree topology. The following table provides a list of port role types and the port and/or bridge properties that the Spanning Tree Algorithm examines to determine which role to assign to the port.

Role	Port/Bridge Properties
Root Port	Port connection that provides the shortest path (lowest path cost value) to the root. The root bridge does not have a root port.
Designated Port	The designated bridge provides the LAN with the shortest path to the root. The designated port connects the LAN to this bridge.
Backup Port	Any operational port on the designated bridge that is not a root or designated port. Provides a backup connection for the designated port. A backup port can only exist when there are redundant designated port connections to the LAN.
Alternate Port	Any operational port that is not the root port for its bridge and its bridge is not the designated bridge for the LAN. An alternate port offers an alternate path to the root bridge if the root port on its own bridge goes down.
Disabled Port	Port is not operational. If an active connection does come up on the port, it is assigned an appropriate role.

Note. The distinction between a backup port and an alternate port was introduced with the IEEE 802.1w standard to help define rapid transition of an alternate port to a root port.

The role a port plays or may potentially play in the active Spanning Tree topology determines the port's operating state; *discarding, learning* or *forwarding*. The *port state* is also configurable in that it is possible to enable or disable a port's administrative status and/or specify a forwarding or blocking state that is only changed through user intervention.

The Spanning Tree Algorithm only includes ports in its calculations that are operational (link is up) and have an enabled administrative status. The following table compares and defines 802.1D and 802.1w port states and their associated port roles:

STP Port State	RSTP Port State	Port State Definition	Port Role
Disabled	Discarding	Port is down or administratively disabled and is not included in the topology.	Disabled
Blocking	Discarding	Frames are dropped, nothing is learned or forwarded on the port. Port is temporarily excluded from topology.	Alternate, Backup
Listening	Discarding	Port is preparing to transmit data and is included in the active topology.	Root, Designated
Learning	Learning	Port is learning MAC addresses that are seen on the port and adding them to the bridge forwarding table, but not transmitting any data. Port is included in the active topology.	Root, Designated
Forwarding	Forwarding	Port is transmitting and receiving data and is included in the active topology.	Root, Designated

Once the Spanning Tree is calculated, there is only one root bridge, one designated bridge for each LAN, and one root port on each bridge (except for the root bridge). Data travels back and forth between bridges over forwarding port connections that form the best, non-redundant path to the root. The active topology ensures that network loops do not exist.

Bridge Protocol Data Units (BPDU)

Switches send layer 2 frames, referred to as Configuration Bridge Protocol Data Units (BPDU), to relay information to other switches. The information in these BPDU is used to calculate and reconfigure the Spanning Tree topology. A Configuration BPDU contains the following information that pertains to the bridge transmitting the BPDU:

Root ID	The Bridge ID for the bridge that this bridge believes is the root.
Root Path Cost	The sum of the Path Costs that lead from the root bridge to this bridge port. The Path Cost is a configurable parameter value. The IEEE 802.1D standard specifies a default value that is based on port speed. See "Configuring Port Path Cost" on page 6-25 for more information.
Bridge ID	An eight-byte hex value that identifies this bridge within the Spanning Tree. The first two bytes contain a configurable priority value and the remaining six bytes contain a bridge MAC address. See "Configuring the Bridge Priority" on page 6-16 for more information. Each switch chassis is assigned a dedicated base MAC address. This is the MAC address that is combined with the priority value to provide a unique Bridge ID for the switch. For more information about the base MAC address, see the *OmniSwitch 7700/7800/8800 Hardware Users Guide*.
Port ID	A 16-bit hex value that identifies the bridge port that transmitted this BPDU. The first 4 bits contain a configurable priority value and the remaining 12 bits contain the physical switch port number. See "Configuring Port Priority" on page 6-24 for more information.

The sending and receiving of Configuration BPDU between switches participating in the bridged network is how the root bridge is elected and the best path to the root is determined and then advertised to the rest of the network. BPDU provide enough information for the STP software running on each switch to determine the following:

- Which bridge will serve as the root bridge.
- The shortest path between each bridge and the root bridge.
- Which bridge will serve as the designated bridge for the LAN.
- Which port on each bridge will serve as the root port.
- The port state (forwarding or discarding) for each bridge port based on the role the port will play in the active Spanning Tree topology.

The following events trigger the transmitting and/or processing of BPDU in order to discover and maintain the Spanning Tree topology.

- When a bridge first comes up, it assumes it is the root and starts transmitting Configuration BPDU on all its active ports advertising its own bridge ID as the root bridge ID.

- When a bridge receives BPDU on its root port that contains more attractive information (higher priority parameters and/or lower path costs), it forwards this information on to other LANs to which it is connected for consideration.

- When a bridge receives BPDU on its designated port that contains information that is less attractive (lower priority values and/or higher path costs), it forwards its own information to other LANs to which it is connected for consideration.

STP evaluates BPDU parameter values to select the best BPDU based on the following order of precedence:

1. The lowest root bridge ID (lowest priority value, then lowest MAC address).

2. The best root path cost.

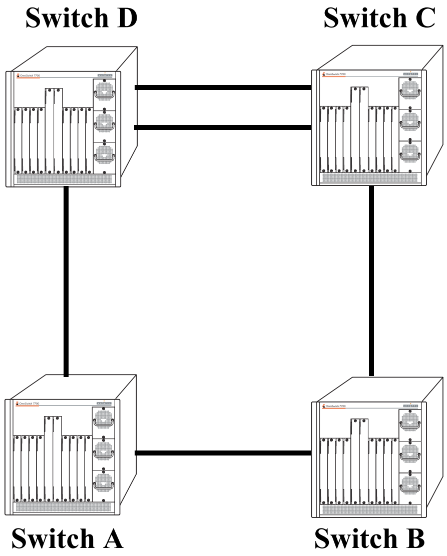
3. If root path costs are equal, the bridge ID of the bridge sending the BPDU.

4. If the previous three values tie, then the port ID (lowest priority value, then lowest port number).

When a topology change occurs, such as when a link goes down or a switch is added to the network, the affected bridge sends Topology Change Notification (TCN) BPDU to the designated bridge for its LAN. The designated bridge will then forward the TCN to the root bridge. The root then sends out a Configuration BPDU and sets a Topology Change (TC) flag within the BPDU to notify other bridges that there is a change in the configuration information. Once this change is propagated throughout the Spanning Tree network, the root stops sending BPDU with the TC flag set and the Spanning Tree returns to an active, stable topology.

Topology Examples

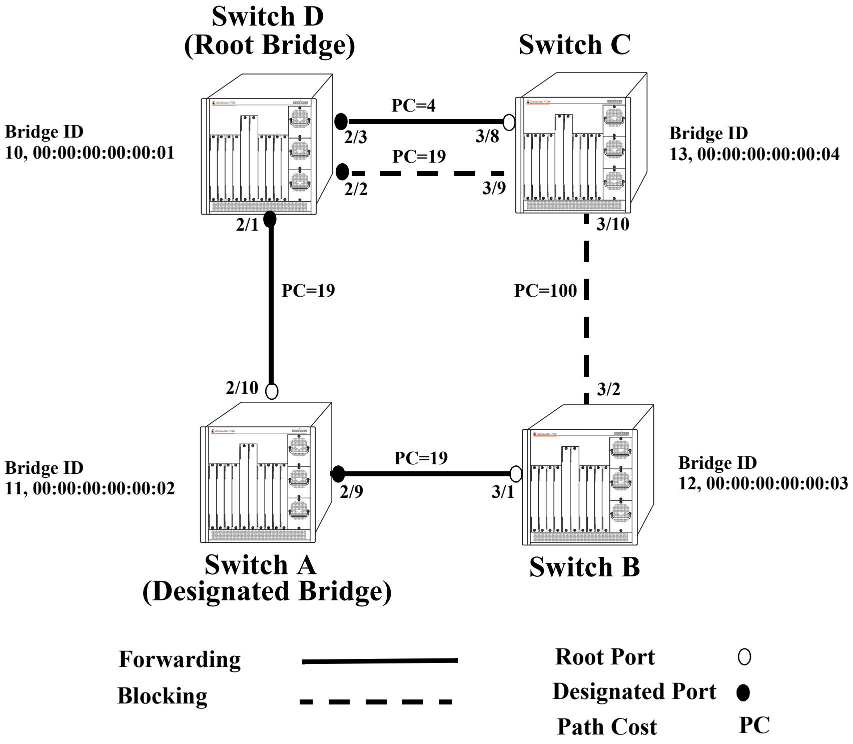
The following diagram shows an example of a physical network topology that incorporates data path redundancy to ensure fault tolerance. These redundant paths, however, create loops in the network configuration. If a device connected to Switch A sends broadcast packets, Switch A will flood the packets out all of its active ports. The switches connected to Switch A will in turn flood the broadcast packets out their active ports, and Switch A will eventually receive the same packets back and the cycle will start over again. This causes severe congestion on the network, often referred to as a *broadcast storm*.



Physical Topology Example

The Spanning Tree Algorithm prevents network loops by ensuring that there is always only one active link between any two switches. This is done by transitioning one of the redundant links into a blocking state, leaving only one link actively forwarding traffic. If the active link goes down, then Spanning Tree will transition one of the blocked links to the forwarding state to take over for the downed link. If a new switch is added to the network, the Spanning Tree topology is automatically recalculated to include the monitoring of links to the new switch.

The following diagram shows the logical connectivity of the same physical topology as determined by the Spanning Tree Algorithm.



Active Spanning Tree Topology Example

In the above active Spanning Tree topology example, the following configuration decisions were made as a result of calculations performed by the Spanning Tree Algorithm:

- Switch D is the root bridge because its bridge ID has a priority value of 10 (the lower the priority value, the higher the priority the bridge has in the Spanning Tree). If all four switches had the same priority, then the switch with the lowest MAC address in its bridge ID would become the root.
- Switch A is the designated bridge for Switch B, because it provides the best path for Switch B to the root bridge.
- Port 2/9 on Switch A is a designated port, because it connects the LAN from Switch B to Switch A.
- All ports on Switch D are designated ports, because Switch D is the root and each port connects to a LAN.
- Ports 2/10, 3/1, and 3/8 are the root ports for Switches A, B, and C, respectively, because they offer the shortest path towards the root bridge.
- The port 3/9 connection on Switch C to port 2/2 on Switch D is in a discarding (blocking) state, as the connection these ports provides is redundant (backup) and has a higher path cost value than the 2/3 to 3/8 connection between the same two switches. As a result, a network loop is avoided.
- The port 3/2 connection on Switch B to port 3/10 on Switch C is also in a discarding (blocking) state, as the connection these ports provides has a higher path cost to root Switch D than the path between Switch B and Switch A. As a result, a network loop is avoided.

Spanning Tree Operating Modes

The switch can operate in one of two Spanning Tree modes: *flat* and *1x1*. Both modes apply to the entire switch and determine whether a single Spanning Tree instance is applied across multiple VLANs (flat mode) or a single instance is applied to each VLAN (1x1 mode). By default, a switch is running in the 1x1 mode when it is first turned on.

Use the **bridge mode** command to select the flat or 1x1 Spanning Tree mode.The switch operates in one mode or the other, however, it is not necessary to reboot the switch when changing modes. To determine which mode the switch is operating in, use the **show spantree** command. For more information about this command, see the *OmniSwitch CLI Reference Guide*.

Using the Flat Spanning Tree Mode

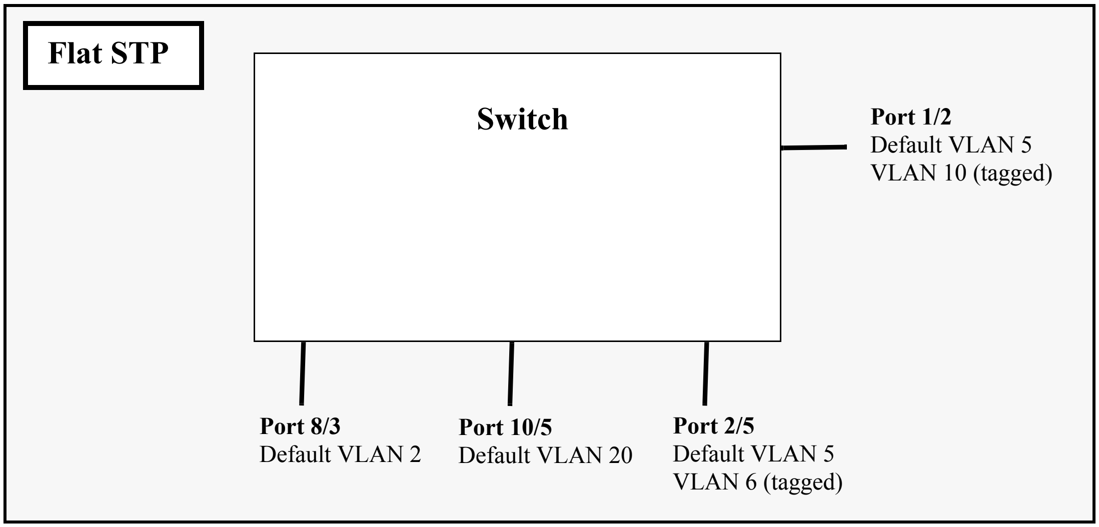
Before selecting the flat Spanning Tree mode, consider the following:

- If STP (802.1D) is the active protocol, then there is one Spanning Tree instance for the entire switch; port states are determined across VLANs. If MSTP (802.1s) is the active protocol, then multiple instances up to a total of 17 are allowed. Port states, however, are still determined across VLANs.
- Multiple connections between switches are considered redundant paths even if they are associated with different VLANs.
- Spanning Tree parameters are configured for the single flat mode instance. For example, if Spanning Tree is disabled on VLAN 1, then it is disabled for all VLANs. Disabling STP on any other VLAN, however, only exclude ports associated with that VLAN from the Spanning Tree Algorithm.
- Fixed (untagged) and 802.1Q tagged ports are supported in each VLAN. BPDU, however, are always untagged.
- When the Spanning Tree mode is changed from 1x1 to flat, ports still retain their VLAN associations but are now part of a single Spanning Tree instance that spans across all VLANs. As a result, a path that was forwarding traffic in the 1x1 mode may transition to a blocking state after the mode is changed to flat.

To change the Spanning Tree operating mode to flat, enter the following command.

```
-> bridge mode flat
```

The following diagram shows a flat mode switch with STP (802.1D) as the active protocol. All ports, regardless of their default VLAN configuration or tagged VLAN assignments, are considered part of one Spanning Tree instance. To see an example of a flat mode switch with MSTP (802.1s) as the active protocol, see Chapter 3, “Using 802.1s Multiple Spanning Tree.”



Flat Spanning Tree Example

In the above example, if port 8/3 connects to another switch and port 10/5 connects to that same switch, the Spanning Tree Algorithm would detect a redundant path and transition one of the ports into a blocking state. The same holds true for the tagged ports.

Using 1x1 Spanning Tree Mode

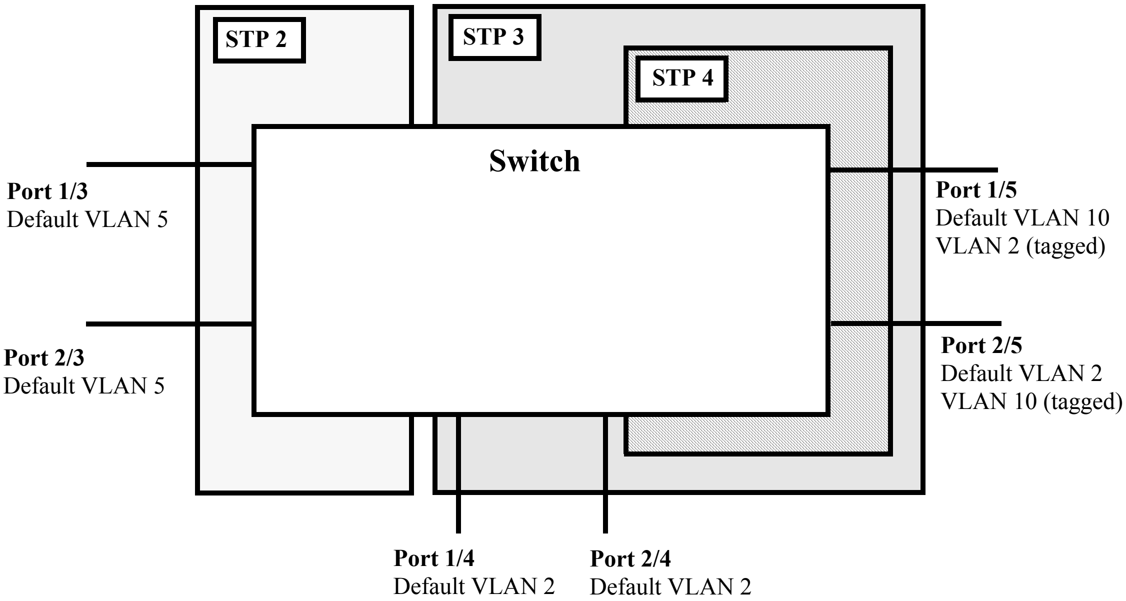
Before selecting the 1x1 Spanning Tree operating mode, consider the following:

- A single Spanning Tree instance is enabled for each VLAN configured on the switch. For example, if there are five VLANs configured on the switch, then there are five separate Spanning Tree instances, each with its own root VLAN. In essence, a VLAN is a virtual bridge in that it will have its own bridge ID and configurable Spanning Tree bridge parameters, such as protocol, priority, hello time, max age, and forward delay.
- Port state is determined on a per VLAN basis. For example, port connections in VLAN 10 are only examined for redundancy within VLAN 10 across all switches. If a port in VLAN 10 and a port in VLAN 20 both connect to the same switch within their respective VLANs, they are not considered redundant data paths and STP will not block one of them. However, if two ports within VLAN 10 both connect to the same switch, then STP will transition one of these ports to a blocking state.
- Fixed (untagged) ports participate in the single Spanning Tree instance that applies to their configured default VLAN.
- 802.1Q tagged ports participate in an 802.1Q Spanning Tree instance that allows the Spanning Tree to extend across tagged VLANs. As a result, a tagged port may participate in more than one Spanning Tree instance; one for each VLAN that the port carries.
- If a VLAN contains both fixed and tagged ports, then a hybrid of the two Spanning Tree instances (single and 802.1Q) is applied. If a VLAN appears as a tag on a port, then the BPDU for that VLAN are also tagged. However, if a VLAN appears as the configured default VLAN for the port, then BPDU are not tagged and the single Spanning Tree instance applies.

To change the Spanning Tree operating mode to 1x1, enter the following command:

```
-> bridge mode 1x1
```

The following diagram shows a switch running in the 1x1 Spanning Tree mode and shows Spanning Tree participation for both fixed and tagged ports.



1x1 (single and 802.1Q) Spanning Tree Example

In the above example, STP2 is a single Spanning Tree instance since VLAN 5 contains only fixed ports. STP 3 and STP 4 are a combination of single and 802.1Q Spanning Tree instances because VLAN 2 contains both fixed and tagged ports. On ports where VLAN 2 is the default VLAN, BPDU are not tagged. On ports where VLAN 2 is a tagged VLAN, BPDU are also tagged.

Configuring Spanning Tree Bridge Parameters

The Spanning Tree software is active on all switches by default and uses default bridge and port parameter values to calculate a loop free topology. It is only necessary to configure these parameter values to change how the topology is calculated and maintained.

Note the following when configuring Spanning Tree bridge parameters:

- When a switch is running in the 1x1 Spanning Tree mode, each VLAN is in essence a virtual bridge with its own Spanning Tree instance and configurable bridge parameters.
- When the switch is running in the flat mode and STP (802.1D) or RSTP (802.1w) is the active protocol, bridge parameter values are only configured for the flat mode instance.
- If MSTP (802.1s) is the active protocol, then the priority value is configurable for each Multiple Spanning Tree Instance (MSTI). All other parameters, however, are still only configured for the flat mode instance and are applied across all MSTIs.
- Bridge parameter values for a VLAN instance are not active unless Spanning Tree is enabled on the VLAN and at least one active port is assigned to the VLAN. Use the **vlan stp** command to enable or disable a VLAN Spanning Tree instance.
- If Spanning Tree is disabled on a VLAN, active ports associated with that VLAN are excluded from Spanning Tree calculations and will remain in a forwarding state.
- Note that when a switch is running in the flat mode, disabling Spanning Tree on VLAN 1 disables the instance for all VLANs and all active ports are then excluded from any Spanning Tree calculations and will remain in a forwarding state.

To view current Spanning Tree bridge parameter values, use the **show spantree** command. For more information about this command, see the *OmniSwitch CLI Reference Guide*.

Bridge Configuration Commands Overview

Spanning Tree bridge commands are available in an implicit form and an explicit form. Implicit commands resemble commands that were previously released with this feature. The type of instance configured with these commands is determined by the Spanning Tree operating mode that is active at the time the command is used. For example, if the 1x1 mode is active, the instance number specified with the command implies a VLAN ID. If the flat mode is active, the single flat mode instance is implied and thus configured by the command.

Explicit commands introduce three new keywords: **cist**, **1x1**, and **msti**. Each of these keywords when used with a bridge command explicitly identify the type of instance that the command will configure. As a result, explicit commands only configure the type of instance identified by the explicit keyword regardless of which mode (1x1 or flat) is active.

The **cist** keyword specifies the Common and Internal Spanning Tree (CIST) instance. The CIST is the single Spanning Tree flat mode instance that is available on all switches. When using STP or RSTP, the CIST is also known as instance 1 or bridge 1. When using MSTP (802.1s), the CIST is also known as instance 0. In either case, an instance number is not required with **cist** commands, as there is only one CIST instance.

The **1x1** keyword indicates that the instance number specified with the command is a VLAN ID. The **msti** keyword indicates that the instance number specified with the command is an 802.1s Multiple Spanning Tree Instance (MSTI).

Note that explicit commands using the **cist** and **msti** keywords are required to define an MSTP (802.1s) configuration. Implicit commands are only allowed for defining STP or RSTP configurations. See Chapter 3, "Using 802.1s Multiple Spanning Tree," for more information about these keywords and using implicit and explicit commands.

The following is a summary of Spanning Tree bridge configuration commands. For more information about these commands, see the *OmniSwitch CLI Reference Guide.*

Commands	Type	Used for ...
bridge protocol	Implicit	Configuring the protocol for a VLAN instance when the 1x1 mode is active or the single Spanning Tree instance when the flat mode is active.
bridge cist protocol	Explicit	Configuring the protocol for the single flat mode instance.
bridge 1x1 protocol	Explicit	Configuring the protocol for a VLAN instance.
bridge priority	Implicit	Configuring the priority value for a VLAN instance or the flat mode instance.
bridge cist priority	Explicit	Configuring the priority value for the single flat mode instance.
bridge msti priority	Explicit	Configuring the protocol for an 802.1s Multiple Spanning Tree Instance (MSTI).
bridge 1x1 priority	Explicit	Configuring the priority value for a VLAN instance.
bridge hello time	Implicit	Configuring the hello time value for a VLAN instance when the 1x1 mode is active or the single Spanning Tree instance when the flat mode is active.
bridge cist hello time	Explicit	Configuring the hello time value for the single flat mode instance.
bridge 1x1 hello time	Explicit	Configuring the hello time value for a VLAN instance.
bridge max age	Implicit	Configuring the maximum age time value for a VLAN instance when the 1x1 mode is active or the single Spanning Tree instance when the flat mode is active.
bridge cist max age	Explicit	Configuring the maximum age time value for the single flat mode instance.
bridge 1x1 max age	Explicit	Configuring the maximum age time value for a VLAN instance.
bridge forward delay	Implicit	Configuring the forward delay time value for a VLAN instance when the 1x1 mode is active or the single Spanning Tree instance when the flat mode is active.
bridge cist forward delay	Explicit	Configuring the forward delay time value for the single flat mode instance.
bridge 1x1 forward delay	Explicit	Configuring the forward delay time value for a VLAN instance.
bridge bpdu-switching	N/A	Configuring the BPDU switching status for a VLAN.

The following sections provide information and procedures for using implicit bridge configuration commands and also includes explicit command examples.

Note. When a snapshot is taken of the switch configuration, the explicit form of all Spanning Tree commands is captured. For example, if the bridge protocol for the flat mode instance was changed from STP to MSTP, then **bridge cist protocol mstp** is the command syntax captured to reflect this in the snapshot file. In addition, explicit commands are captured for both flat and 1x1 mode configurations.

Selecting the Bridge Protocol

The switch supports three Spanning Tree protocols: 802.1D (STP), 802.1w (RSTP), and 802.1s (MSTP). By default, 802.1D is the protocol used for all instances.

To configure the Spanning Tree protocol for a VLAN instance when the switch is running in the 1x1 mode, enter **bridge** followed by an existing VLAN ID, then **protocol** followed by **stp** or **rstp**. For example, the following command changes the protocol to RSTP for VLAN 455:

```
-> bridge 455 protocol rstp
```

Note that when configuring the protocol value for a VLAN instance, MSTP is not an available option. This protocol is only supported on the flat mode instance.

In addition, the explicit **bridge 1x1 protocol** command configures the protocol for a VLAN instance regardless of which mode (1x1 or flat) is active on the switch. For example, the following command also changes the protocol for VLAN 455 to RSTP:

```
-> bridge 1x1 455 protocol rstp
```

To configure the protocol for the single flat mode instance when the switch is running in either mode (1x1 or flat), use the **bridge protocol** command but do *not* specify an instance number. This command configures the flat mode instance by default, so an instance number is not needed, as shown in the following example:

```
-> bridge protocol mstp
```

As in previous releases, it is possible to configure the flat mode instance with the **bridge protocol** command by specifying **1** as the instance number (e.g., **bridge 1 protocol rstp**). However, this is only available when the switch is already running in the flat mode and STP or RSTP is the active protocol.

In addition, the explicit **bridge cist protocol** command configures the protocol for the flat mode instance regardless of which mode (1x1 or flat) is active on the switch. For example, the following command selects the RSTP protocol for the flat mode instance:

```
-> bridge cist protocol mstp
```

Configuring the Bridge Priority

A bridge is identified within the Spanning Tree by its bridge ID (an eight byte hex number). The first two bytes of the bridge ID contain a priority value and the remaining six bytes contain a bridge MAC address.

The bridge priority is used to determine which bridge will serve as the root of the Spanning Tree. The lower the priority value, the higher the priority. If more than one bridge have the same priority, then the bridge with the lowest MAC address becomes the root.

Note. Configuring a Spanning Tree bridge instance with a priority value that will cause the instance to become the root is recommended, instead of relying on the comparison of switch base MAC addresses to determine the root.

If the switch is running in the 1x1 Spanning Tree mode, then a priority value is assigned to each VLAN instance. If the switch is running in the flat Spanning Tree mode, the priority is assigned to the flat mode instance or an 802.1s Multiple Spanning Tree Instance (MSTI). In both cases, the default priority value assigned is 32768. Note that priority values for an MSTI must be multiples of 4096.

To change the bridge priority value for a VLAN instance, specify a VLAN ID with the **bridge priority** command when the switch is running in the 1x1 mode. For example, the following command changes the priority for VLAN 455 to 25590:

```
-> bridge 455 priority 25590
```

The explicit **bridge 1x1 priority** command configures the priority for a VLAN instance when the switch is running in either mode (1x1 or flat). For example, the following command performs the same function as the command in the previous example:

```
-> bridge 1x1 455 priority 25590
```

To change the bridge priority value for the flat mode instance, use either the **bridge priority** command or the **bridge cist priority** command. Note that both commands are available when the switch is running in either mode (1x1 or flat) and an instance number is not required. For example, the following commands change the priority value for the flat mode instance to 12288:

```
-> bridge priority 12288
-> bridge cist priority 12288
```

As in previous releases, it is possible to configure the flat mode instance with the **bridge protocol** command by specifying **1** as the instance number (e.g., **bridge 1 protocol rstp**). However, this is only available when the switch is already running in the flat mode and STP or RSTP is the active protocol.

The bridge priority value is also configurable for an 802.1s Multiple Spanning Tree Instance (MSTI). To configure this value for an MSTI, use the explicit **bridge msti priority** command and specify the MSTI ID for the instance number and a priority value that is a multiple of 4096. For example, the following command configures the priority value for MSTI 10 to 61440:

```
-> bridge msti 10 priority 61440
```

Note that when MSTP (802.1s) is the active flat mode protocol, explicit Spanning Tree bridge commands are required to configure parameter values. Implicit commands are for configuring parameters when the STP or RSTP protocols are in use. See Chapter 3, “Using 802.1s Multiple Spanning Tree,” for more information.

Configuring the Bridge Hello Time

The bridge hello time interval is the number of seconds a bridge will wait between transmissions of Configuration BPDU. When a bridge is attempting to become the root or if it has become the root or a designated bridge, it sends Configuration BPDU out all forwarding ports once every hello time value.

The hello time propagated in a root bridge Configuration BPDU is the value used by all other bridges in the tree for their own hello time. Therefore, if this value is changed for the root bridge, all other bridges associated with the same STP instance will adopt this value as well.

Note that lowering the hello time interval improves the robustness of the Spanning Tree algorithm. Increasing the hello time interval lowers the overhead of Spanning Tree processing.

If the switch is running in the 1x1 Spanning Tree mode, then a hello time value is defined for each VLAN instance. If the switch is running in the flat Spanning Tree mode, then a hello time value is defined for the single flat mode instance. In both cases, the default hello time value used is 2 seconds.

To change the bridge hello time value for a VLAN instance, specify a VLAN ID with the **bridge hello time** command when the switch is running in the 1x1 mode. For example, the following command changes the hello time for VLAN 455 to 5 seconds:

```
-> bridge 455 hello time 5
```

The explicit **bridge 1x1 hello time** command configures the hello time value for a VLAN instance when the switch is running in either mode (1x1 or flat). For example, the following command performs the same function as the command in the previous example:

```
-> bridge 1x1 455 hello time 5
```

To change the bridge hello time value for the flat mode instance, use either the **bridge hello time** command or the **bridge cist hello time** command. Note that both commands are available when the switch is running in either mode (1x1 or flat) and an instance number is not required. For example, the following commands change the hello time value for the flat mode instance to 12288:

```
-> bridge hello time 10
-> bridge cist hello time 10
```

As in previous releases, it is possible to configure the flat mode instance with the **bridge hello time** command by specifying **1** as the instance number (e.g., **bridge 1 hello time 5**). However, this is only available when the switch is already running in the flat mode and STP or RSTP is the active protocol.

Note that the bridge hello time is not configurable for 802.1s Multiple Spanning Tree Instances (MSTI). These instances inherit the hello time from the flat mode instance (CIST).

Configuring the Bridge Max Age Time

The bridge max age time specifies how long, in seconds, the bridge retains Spanning Tree information it receives from Configuration BPDU. When a bridge receives a BPDU, it updates its configuration information and the max age timer is reset. If the max age timer expires before the next BPDU is received, the bridge will attempt to become the root, designated bridge, or change its root port.

The max age time propagated in a root bridge Configuration BPDU is the value used by all other bridges in the tree for their own max age time. Therefore, if this value is changed for the root bridge, all other VLANs associated with the same instance will adopt this value as well.

If the switch is running in the 1x1 Spanning Tree mode, then a max age time value is defined for each VLAN instance. If the switch is running in the flat Spanning Tree mode, then the max age value is defined for the flat mode instance. In both cases, the default max age time used is 20 seconds.

Note that configuring a low max age time may cause Spanning Tree to reconfigure the topology more often.

To change the bridge max age time value for a VLAN instance, specify a VLAN ID with the **bridge max age** command when the switch is running in the 1x1 mode. For example, the following command changes the max age time for VLAN 455 to 10 seconds:

```
-> bridge 455 max age 10
```

The explicit **bridge 1x1 max age** command configures the max age time for a VLAN instance when the switch is running in either mode (1x1 or flat). For example, the following command performs the same function as the command in the previous example:

```
-> bridge 1x1 455 max age 10
```

To change the max age time value for the flat mode instance, use either the **bridge max age** command or the **bridge cist max age** command. Note that both commands are available when the switch is running in either mode (1x1 or flat) and an instance number is not required. For example, the following commands change the max age time for the flat mode instance to 10:

```
-> bridge max age 10
-> bridge cist max age 10
```

As in previous releases, it is possible to configure the flat mode instance with the **bridge max age** command by specifying **1** as the instance number (e.g., **bridge 1 max age 30**). However, this is only available when the switch is already running in the flat mode and STP or RSTP is the active protocol.

Note that the max age time is not configurable for 802.1s Multiple Spanning Tree Instances (MSTI). These instances inherit the max age time from the flat mode instance (CIST).

Configuring the Bridge Forward Delay Time

The bridge forward delay time specifies how long, in seconds, a port remains in the learning state while it is transitioning to a forwarding state. In addition, when a topology change occurs, the forward delay time value is used to age out all dynamically learned addresses in the MAC address forwarding table. For more information about the MAC address table, see Chapter 2, “Managing Source Learning.”

The forward delay time propagated in a root bridge Configuration BPDU is the value used by all other bridges in the tree for their own forward delay time. Therefore, if this value is changed for the root bridge, all other bridges associated with the same instance will adopt this value as well.

If the switch is running in the 1x1 Spanning Tree mode, then a forward delay time value is defined for each VLAN instance. If the switch is running in the flat Spanning Tree mode, then the forward delay time value is defined for the flat mode instance. In both cases, the default forward delay time used is 15 seconds.

Note that specifying a low forward delay time may cause temporary network loops, because packets may get forwarded before Spanning Tree configuration or change notices have reached all nodes in the network.

To change the bridge forward delay time value for a VLAN instance, specify a VLAN ID with the **bridge forward delay** command when the switch is running in the 1x1 mode. For example, the following command changes the forward delay time for VLAN 455 to 10 seconds:

```
> bridge 455 forward delay 20
```

The explicit **bridge 1x1 forward delay** command configures the forward delay time for a VLAN instance when the switch is running in either mode (1x1 or flat). For example, the following command performs the same function as the command in the previous example:

```
-> bridge 1x1 455 forward delay 20
```

To change the forward delay time value for the flat mode instance, use either the **bridge forward delay** command or the **bridge cist forward delay** command. Note that both commands are available when the switch is running in either mode (1x1 or flat) and an instance number is not required. For example, the following commands change the forward delay time for the flat mode instance to 10:

```
-> bridge forward delay 10
-> bridge cist forward delay 10
```

As in previous releases, it is possible to configure the flat mode instance with the **bridge forward delay** command by specifying **1** as the instance number (e.g., **bridge 1 forward delay 30**). However, this is only available when the switch is already running in the flat mode and STP or RSTP is the active protocol.

Note that the forward delay time is not configurable for 802.1s Multiple Spanning Tree Instances (MSTI). These instances inherit the forward delay time from the flat mode instance (CIST).

Enabling/Disabling the VLAN BPDU Switching Status

By default, BPDU are not switched on ports associated with VLANs that have Spanning Tree disabled. This may result in a network loop if the VLAN has redundant paths to one or more other switches. Allowing VLANs that have Spanning Tree disabled to forward BPDU to all ports in the VLAN, can help to avoid this problem.

To enable or disable BPDU switching on a VLAN, enter **bridge** followed by an existing VLAN ID (or VLAN 1 if using a flat Spanning Tree instance) then **bpdu-switching** followed by **enable** or **disable**. For example, the following commands enable BPDU switching on VLAN 10 and disable it on VLAN 20:

```
-> bridge 10 bpdu-switching enable
-> bridge 20 bpdu-switching disable
```

Note. Make sure that disabling BPDU switching on a Spanning Tree disabled VLAN will not cause network loops to go undetected.

Configuring the Path Cost Mode

The path cost mode controls whether the switch uses a 16-bit port path cost (PPC) or a 32-bit PPC. When a 32-bit PPC switch connects to a 16-bit PPC switch, the 32-bit switch will have a higher PPC value that will advertise an inferior path cost to the 16-bit switch. In this case, it may be desirable to set the 32-bit switch to use STP or RSTP with a 16-bit PPC value.

By default, the path cost mode is set to automatically use a 16-bit value for all ports that are associated with an STP (802.1D) instance or an RSTP (802.1w) instance and a 32-bit value for all ports associated with an MSTP (802.1s) value. It is also possible to set the path cost mode to always use a 32-bit regardless of which protocol is active.

To change the path cost mode, use the **bridge path cost mode** command and specify either **auto** (uses PPC value based on protocol) or **32bit**.(always use a 32-bit PPC value). For example, the following command changes the default path cost mode, which is automatic, to 32-bit mode:

```
-> bridge path cost mode 32bit
```

Configuring Spanning Tree Port Parameters

The following sections provide information and procedures for using CLI commands to configure STP port parameters. These parameters determine the behavior of a port for a specific VLAN Spanning Tree instance (1x1 STP mode) or for a single Spanning Tree instance applied to the entire switch (flat STP mode).

When a switch is running in the 1x1 STP mode, each VLAN is in essence a virtual STP bridge with its own STP instance and configurable parameters. To change STP port parameters while running in this mode, a VLAN ID is specified to identify the VLAN STP instance associated with the specified port. When a switch is running in the flat Spanning Tree mode, VLAN 1 is specified for the VLAN ID. It is possible to configure STP parameters on other VLANs while running in this mode, but only VLAN 1 parameter values apply to all Spanning Tree ports.

Only bridged ports participate in the Spanning Tree Algorithm. A port is considered bridged if it meets all the following criteria:

- Port is either a fixed (non-mobile) port, an 802.1Q tagged port or a link aggregate logical port.
- Spanning tree is enabled on the port.
- Port is assigned to a VLAN that has Spanning Tree enabled.
- Port state (forwarding or blocking) is dynamically determined by the Spanning Tree Algorithm, not manually set.

Bridge Configuration Commands Overview

Spanning Tree port commands are available in an implicit form and an explicit form. Implicit commands resemble commands that were previously released with this feature. The type of instance configured with these commands is determined by the Spanning Tree operating mode that is active at the time the command is used. For example, if the 1x1 mode is active, the instance number specified with the command implies a VLAN ID. If the flat mode is active, the single flat mode instance is implied and thus configured by the command.

Explicit commands introduce three new keywords: **cist**, **1x1**, and **msti**. Each of these keywords when used with a port command explicitly identify the type of instance that the command will configure. As a result, explicit commands only configure the type of instance identified by the explicit keyword regardless of which mode (1x1 or flat) is active.

The **cist** keyword specifies the Common and Internal Spanning Tree (CIST) instance. The CIST is the single Spanning Tree flat mode instance that is available on all switches. When using STP or RSTP, the CIST is also known as instance 1 or bridge 1. When using MSTP (802.1s), the CIST is also known as instance 0. In either case, an instance number is not required with **cist** commands, as there is only one CIST instance.

The **1x1** keyword indicates that the instance number specified with the command is a VLAN ID. The **msti** keyword indicates that the instance number specified with the command is an 802.1s Multiple Spanning Tree Instance (MSTI).

Note that explicit commands using the **cist** and **msti** keywords are required to define an MSTP (802.1s) configuration. Implicit commands are only allowed for defining STP or RSTP configurations. See Chapter 3, “Using 802.1s Multiple Spanning Tree,” for more information about these keywords and using implicit and explicit commands.

The following is a summary of Spanning Tree port configuration commands. For more information about these commands, see the *OmniSwitch CLI Reference Guide.*

Commands	Type	Used for ...
bridge slot/port	Implicit	Configuring the port Spanning Tree status for a VLAN instance when the 1x1 mode is active or the single Spanning Tree instance when the flat mode is active.
bridge cist port	Explicit	Configuring the port Spanning Tree status for the single flat mode instance.
bridge 1x1 port	Explicit	Configuring the port Spanning Tree status for a VLAN instance.
bridge slot/port priority	Implicit	Configuring the port priority value for a VLAN instance when the 1x1 mode is active or the single Spanning Tree instance when the flat mode is active.
bridge cist slot/port priority	Explicit	Configuring the port priority value for the single flat mode instance.
bridge msti slot/port priority	Explicit	Configuring the port priority value for an 802.1s Multiple Spanning Tree Instance (MSTI).
bridge 1x1 slot/port priority	Explicit	Configuring the port priority value for a VLAN instance.
bridge slot/port path cost	Implicit	Configuring the port path cost value for a VLAN instance when the 1x1 mode is active or the single Spanning Tree instance when the flat mode is active.
bridge cist slot/port path cost	Explicit	Configuring the port path cost value for the single flat mode instance.
bridge msti slot/port path cost	Explicit	Configuring the port path cost value for an 802.1s Multiple Spanning Tree Instance (MSTI).
bridge 1x1 slot/port path cost	Explicit	Configuring the port path cost value for a VLAN instance.
bridge slot/port mode	Explicit	Configuring the port Spanning Tree mode (dynamic or manual) for a VLAN instance when the 1x1 mode is active or the single Spanning Tree instance when the flat mode is active.
bridge cist slot/port mode	Implicit	Configuring the port Spanning Tree mode (dynamic or manual) for the single flat mode instance.
bridge 1x1 slot/port mode	Explicit	Configuring the port Spanning Tree mode (dynamic or manual) for a VLAN instance.
bridge slot/port connection	Explicit	Configuring the port connection type for a VLAN instance when the 1x1 mode is active or the single Spanning Tree instance when the flat mode is active.
bridge cist slot/port connection	Implicit	Configuring the port connection type for the single flat mode instance.
bridge 1x1 slot/port connection	Explicit	Configuring the port connection type for a VLAN instance.

The following sections provide information and procedures for using implicit Spanning Tree port configuration commands and also includes explicit command examples.

Note. When a snapshot is taken of the switch configuration, the explicit form of all Spanning Tree commands is captured. For example, if the bridge protocol for the flat mode instance was changed from STP to MSTP, then **bridge cist protocol mstp** is the command syntax captured to reflect this in the snapshot file. In addition, explicit commands are captured for both flat and 1x1 mode configurations.

Enabling/Disabling Spanning Tree on a Port

By default, Spanning Tree is enabled on all ports. When Spanning Tree is disabled on a port, the port is put in a forwarding state for the specified instance. For example, if a port is associated with both VLAN 10 and VLAN 20 and Spanning Tree is disabled on the port for VLAN 20, the port state is set to forwarding for VLAN 20. However, the VLAN 10 instance still controls the port's state as it relates to VLAN 10. This example assumes the switch is running in the 1x1 Spanning Tree mode.

If the switch is running in the flat Spanning Tree mode, then disabling the port Spanning Tree status applies across all VLANs associated with the port. The flat mode instance is specified as the port's instance, even if the port is associated with multiple VLANs.

To change the port Spanning Tree status for a VLAN instance, specify a VLAN ID with the **bridge slot/port** command when the switch is running in the 1x1 mode. For example, the following commands enable Spanning Tree on port 8/1 for VLAN 10 and disable STP on port 6/2 for VLAN 20:

```
-> bridge 10 8/1 enable
-> bridge 20 6/2 disable
```

The explicit **bridge 1x1 port** command configures the priority for a VLAN instance when the switch is running in either mode (1x1 or flat). For example, the following commands perform the same function as the commands in the previous example:

```
-> bridge 1x1 10 8/1 enable
-> bridge 1x1 20 6/2 disable
```

To change the port Spanning Tree status for the flat mode instance, use either the **bridge slot/port** command or the **bridge cist port** command. Note that both commands are available when the switch is running in either mode (1x1 or flat) and an instance number is not required. For example, the following commands disable the Spanning Tree status on port 1/24 for the flat mode instance:

```
-> bridge 1/24 disable
-> bridge cist 1/24 disable
```

As in previous releases, it is possible to configure the flat mode instance with the **bridge slot/port** command by specifying **1** as the instance number (e.g., **bridge 1 1/24 enable**). However, this is only available when the switch is already running in the flat mode and STP or RSTP is the active protocol.

Spanning Tree on Link Aggregate Ports

Physical ports that belong to a link aggregate do not participate in the Spanning Tree Algorithm. Instead, the algorithm is applied to the aggregate logical link (virtual port) that represents a collection of physical ports.

To enable or disable the Spanning Tree status for a link aggregate, use the **bridge slot/port** commands described above but specify a link aggregate control number instead of a slot and port. For example, the following command disables Spanning Tree for link aggregate 10 associated with VLAN 755:

```
-> bridge 755 10 disable
```

For more information about configuring an aggregate of ports, see Chapter 11, "Configuring Static Link Aggregation," and Chapter 12, "Configuring Dynamic Link Aggregation."

Configuring Port Priority

A bridge port is identified within the Spanning Tree by its Port ID (a 16-bit or 32-bit hex number). The first 4 bits of the Port ID contain a priority value and the remaining 12 bits contain the physical switch port number. The port priority is used to determine which port offers the best path to the root when multiple paths have the same path cost. The port with the highest priority (lowest numerical priority value) is selected and the others are put into a blocking state. If the priority values are the same for all ports in the path, then the port with the lowest physical switch port number is selected.

By default, Spanning Tree is enabled on a port and the port priority value is set to 7. If the switch is running in the 1x1 Spanning Tree mode, then the port priority applies to the specified VLAN instance associated with the port. If the switch is running in the flat Spanning Tree mode, then the port priority applies across all VLANs associated with the port. The flat mode instance is specified as the port's instance, even if the port is associated with multiple VLANs.

To change the port priority value for a VLAN instance, specify a VLAN ID with the **bridge slot/port priority** command when the switch is running in the 1x1 mode. For example, the following command sets the priority value for port 8/1 to 3 for the VLAN 10 instance:

```
-> bridge 10 8/1 priority 3
```

The explicit **bridge cist slot/port priority** command configures the port priority value for a VLAN instance when the switch is running in either mode (1x1 or flat). For example, the following command performs the same function as the command in the previous example:

```
-> bridge 1x1 10 8/1 priority 3
```

To change the port priority value for the flat mode instance, use either the **bridge slot/port priority** command or the **bridge cist slot/port priority** command. Note that both commands are available when the switch is running in either mode (1x1 or flat) and an instance number is not required. For example, the following commands change the priority value for port 1/24 for the flat mode instance to 15:

```
-> bridge 1/24 priority 15
-> bridge cist 1/24 priority 10
```

As in previous releases, it is possible to configure the flat mode instance with the **bridge slot/port priority** command by specifying **1** as the instance number (e.g., **bridge 1 1/24 priority 15**). However, this is only available when the switch is already running in the flat mode and STP or RSTP is the active protocol.

The port priority value is also configurable for an 802.1s Multiple Spanning Tree Instance (MSTI). To configure this value for an MSTI, use the explicit **bridge msti slot/port priority** command and specify the

MSTI ID for the instance number. For example, the following command configures the priority value for port 1/12 for MSTI 10 to 5:

```
-> bridge msti 10 1/12 priority 5
```

Note that when MSTP (802.1s) is the active flat mode protocol, explicit Spanning Tree bridge commands are required to configure parameter values. Implicit commands are for configuring parameters when the STP or RSTP protocols are in use. See Chapter 3, “Using 802.1s Multiple Spanning Tree,” for more information.

Port Priority on Link Aggregate Ports

Physical ports that belong to a link aggregate do not participate in the Spanning Tree Algorithm. Instead, the algorithm is applied to the aggregate logical link (virtual port) that represents a collection of physical ports.

To change the port priority for a link aggregate, use the **bridge slot/port priority** commands described above, but specify a link aggregate control number instead of a slot and port. For example, the following command sets the priority for link aggregate 10 associated with VLAN 755 to 9:

```
-> bridge 755 10 priority 9
```

For more information about configuring an aggregate of ports, see Chapter 11, “Configuring Static Link Aggregation,” and Chapter 12, “Configuring Dynamic Link Aggregation.”

Configuring Port Path Cost

The path cost value specifies the contribution of a port to the path cost towards the root bridge that includes the port. The root path cost is the sum of all path costs along this same path and is the value advertised in Configuration BPDU transmitted from active Spanning Tree ports. The lower the cost value, the closer the switch is to the root.

Note that type of path cost value used depends on which path cost mode is active (automatic or 32-bit). If the path cost mode is set to automatic, a 16-bit value is used when STP or RSTP is the active protocol and a 32-bit value is used when MSTP is the active protocol. If the mode is set to 32-bit, then a 32-bit path cost value is used regardless of which protocol is active. See “Configuring the Path Cost Mode” on page 6-19 for more information.

If a 32-bit path cost value is in use and the *path_cost* is set to zero, the following IEEE 802.1s recommended default path cost values based on link speed are used:

Link Speed	IEEE 802.1D Recommended Value
10 MB	2,000,000
100 MB	200,000
1 GB	20,000
10 Gbps	2,000

Is a 16-bit path cost value is in use and the *path_cost* is set to zero, the following IEEE 802.1D recommended default path cost values based on link speed are used:

Link Speed	**IEEE 802.1D Recommended Value**
4 Mbps	250
10 Mbps	100
16 Mbps	62
100 Mbps	19
1 Gbps	4
10 Gbps	2

By default, Spanning Tree is enabled on a port and the path cost is set to zero. If the switch is running in the 1x1 Spanning Tree mode, then the port path cost applies to the specified VLAN instance associated with the port. If the switch is running in the flat Spanning Tree mode, then the port path cost applies across all VLANs associated with the port. The flat mode instance is specified as the port's instance, even if the port is associated with other VLANs.

To change the port path cost value for a VLAN instance, specify a VLAN ID with the **bridge slot/port path cost** command when the switch is running in the 1x1 mode. For example, the following command configures a 16-bit path cost value for port 8/1 for VLAN 10 to 19 (the port speed is 100 MB, 19 is the recommended value).

```
-> bridge 10 8/1 path cost 19
```

The explicit **bridge 1x1 slot/port path cost** command configures the port path cost value for a VLAN instance when the switch is running in either mode (1x1 or flat). For example, the following command performs the same function as the command in the previous example:

```
-> bridge 1x1 10 8/1 path cost 19
```

To change the port path cost value for the flat mode instance, use either the **bridge slot/port path cost** command or the **bridge cist slot/port path cost** command. Note that both commands are available when the switch is running in either mode (1x1 or flat) and an instance number is not required. For example, the following commands configure a 32-bit path cost value for port 1/24 for the flat mode instance to 20,000 (the port speed is 1 GB, 20,000 is the recommended value):

```
-> bridge 1/24 path cost 20000
-> bridge cist 1/24 path cost 20000
```

As in previous releases, it is possible to configure the flat mode instance with the **bridge slot/port path cost** command by specifying **1** as the instance number (e.g., **bridge 1 1/24 path cost 19**). However, this is only available when the switch is already running in the flat mode and STP or RSTP is the active protocol.

The port path cost value is also configurable for an 802.1s Multiple Spanning Tree Instance (MSTI). To configure this value for an MSTI, use the explicit **bridge msti slot/port path cost** command and specify the MSTI ID for the instance number. For example, the following command configures the path cost value for port 1/12 for MSTI 10 to 19:

```
-> bridge msti 10 1/12 path cost 19
```

Note that when MSTP (802.1s) is the active flat mode protocol, explicit Spanning Tree bridge commands are required to configure parameter values. Implicit commands are for configuring parameters when the STP or RSTP protocols are in use. See Chapter 3, "Using 802.1s Multiple Spanning Tree," for more information.

Path Cost for Link Aggregate Ports

Physical ports that belong to a link aggregate do not participate in the Spanning Tree Algorithm. Instead, the algorithm is applied to the aggregate logical link (virtual port) that represents a collection of physical ports. By default, Spanning Tree is enabled on the aggregate logical link and the path cost value is set to zero.

To change the path cost value for a link aggregate, use the **bridge slot/port path cost** commands described above, but specify a link aggregate control number instead of a slot and port. For example, the following command sets the path cost for link aggregate 10 associated with VLAN 755 to 19:

```
-> bridge 755 10 path cost 19
```

For more information about configuring an aggregate of ports, see Chapter 11, "Configuring Static Link Aggregation," and Chapter 12, "Configuring Dynamic Link Aggregation."

Configuring Port Mode

There are two port modes supported: manual and dynamic. Manual mode indicates that the port was set by the user to a forwarding or blocking state. The port will operate in the state selected until the state is manually changed again or the port mode is changed to dynamic. Ports operating in a manual mode state do not participate in the Spanning Tree Algorithm. Dynamic mode indicates that the active Spanning Tree Algorithm will determine port state.

By default, Spanning Tree is enabled on the port and the port operates in the dynamic mode. If the switch is running in the 1x1 Spanning Tree mode, then the port mode applies to the specified VLAN instance associated with the port. If the switch is running in the flat Spanning Tree mode, then the port mode applies across all VLANs associated with the port. The flat mode instance is specified as the port's instance, even if the port is associated with other VLANs.

To change the port Spanning Tree mode for a VLAN instance, specify a VLAN ID with the **bridge slot/port mode** command when the switch is running in the 1x1 mode. For example, the following command sets the mode for port 8/1 for VLAN 10 to forwarding.

```
-> bridge 10 8/1 mode forwarding
```

The explicit **bridge 1x1 slot/port mode** command configures the port mode for a VLAN instance when the switch is running in either mode (1x1 or flat). For example, the following command performs the same function as the command in the previous example:

```
-> bridge 1x1 10 8/1 mode forwarding
```

To change the port Spanning Tree mode for the flat mode instance, use either the **bridge slot/port mode** command or the **bridge cist slot/port mode** command. Note that both commands are available when the switch is running in either mode (1x1 or flat) and an instance number is not required. For example, the following commands configure the Spanning Tree mode on port 1/24 for the flat mode instance:

```
-> bridge 1/24 mode blocking
-> bridge cist 1/24 mode blocking
```

As in previous releases, it is possible to configure the flat mode instance with the **bridge slot/port mode** command by specifying **1** as the instance number (e.g., **bridge 1 1/24 mode dynamic**). However, this is only available when the switch is already running in the flat mode and STP or RSTP is the active protocol.

Mode for Link Aggregate Ports

Physical ports that belong to a link aggregate do not participate in the Spanning Tree Algorithm. Instead, the algorithm is applied to the aggregate logical link (virtual port) that represents a collection of physical ports. To change the port mode for a link aggregate, use the **bridge slot/port mode** commands described above, but specify a link aggregate control number instead of a slot and port. For example, the following command sets the port mode for link aggregate 10 associated with VLAN 755 to blocking:

```
-> bridge 755 10 mode blocking
```

For more information about configuring an aggregate of ports, see Chapter 11, “Configuring Static Link Aggregation,” and Chapter 12, “Configuring Dynamic Link Aggregation.”

Configuring Port Connection Type

Specifying a port connection type is done when using the Rapid Spanning Tree Algorithm and Protocol (RSTP), as defined in the IEEE 802.1w standard. RSTP transitions a port from a blocking state directly to forwarding, bypassing the listening and learning states, to provide a rapid reconfiguration of the Spanning Tree in the event of a path or root bridge failure. Rapid transition of a port state depends on the port’s configurable connection type. These types are defined as follows:

- Point-to-point LAN segment (port connects directly to another switch).
- No point-to-point shared media LAN segment (port connects to multiple switches).
- Edge port (port is at the edge of a bridged LAN, does not receive BPDU and has only one MAC address learned). Edge ports, however, will operationally revert to a point to point or a no point to point connection type if a BPDU is received on the port.

A port is considered connected to a point-to-point LAN segment if the port belongs to a link aggregate of ports, or if auto negotiation determines if the port should run in full duplex mode, or if full duplex mode was administratively set. Otherwise, that port is considered connected to a no point-to-point LAN segment.

Rapid transition of a designated port to forwarding can only occur if the port’s connection type is defined as a point to point or an edge port. Defining a port’s connection type as a point to point or as an edge port makes the port eligible for rapid transition, regardless of what actually connects to the port. However, an alternate port transition to the role of root port is always allowed regardless of the alternate port’s connection type.

Note. Configure ports that will connect to a host (PC, workstation, server, etc.) as edge ports so that these ports will transition directly to a forwarding state and not trigger an unwanted topology change when a device is connected to the port. If a port is configured as a point to point or no point to point connection type, the switch will assume a topology change when this port goes active and will flush and relearn all learned MAC addresses for the port’s assigned VLAN.

By default, Spanning Tree is enabled on the port and the connection type is set to auto point to point. The auto point to point setting determines the connection type based on the operational status of the port.

If the switch is running in the 1x1 Spanning Tree mode, then the connection type applies to the specified VLAN instance associated with the port. If the switch is running in the flat Spanning Tree mode, then the connection type applies across all VLANs associated with the port. The flat mode instance is referenced as the port's instance, even if the port is associated with other VLANs.

To change the port connection type for a VLAN instance, specify a VLAN ID with the **bridge slot/port connection** command when the switch is running in the 1x1 mode. For example, the following command defines an edge port connection type for port 8/1 associated with VLAN 10.

```
-> bridge 10 8/1 connection edgeport
```

The explicit **bridge 1x1 slot/port connection** command configures the connection type for a VLAN instance when the switch is running in either mode (1x1 or flat). For example, the following command performs the same function as the command in the previous example:

```
-> bridge 1x1 10 8/1 connection edgeport
```

To change the port Spanning Tree mode for the flat mode instance, use either the **bridge slot/port connection** command or the **bridge cist slot/port connection** command. Note that both commands are available when the switch is running in either mode (1x1 or flat) and an instance number is not required. For example, the following commands configure the connection type for port 1/24 for the flat mode instance:

```
-> bridge 1/24 connection ptp
-> bridge cist 1/24 connection ptp
```

As in previous releases, it is possible to configure the flat mode instance with the **bridge slot/port connection** command by specifying **1** as the instance number (e.g., **bridge 1 1/24 connection noptp**). However, this is only available when the switch is already running in the flat mode and STP or RSTP is the active protocol.

Note that the **bridge slot/port connection** command only configures one port at a time.

Connection Type on Link Aggregate Ports

Physical ports that belong to a link aggregate do not participate in the Spanning Tree Algorithm. Instead, the algorithm is applied to the aggregate logical link (virtual port) that represents a collection of physical ports. To change the port connection type for a link aggregate, use the **bridge slot/port connection** commands described above, but specify a link aggregate control number instead of a slot and port. For example, the following command defines the link aggregate 10 associated with VLAN 755 as an edge port:

```
-> bridge 755 10 connection edgeport
```

For more information about configuring an aggregate of ports, see Chapter 11, "Configuring Static Link Aggregation," and Chapter 12, "Configuring Dynamic Link Aggregation."

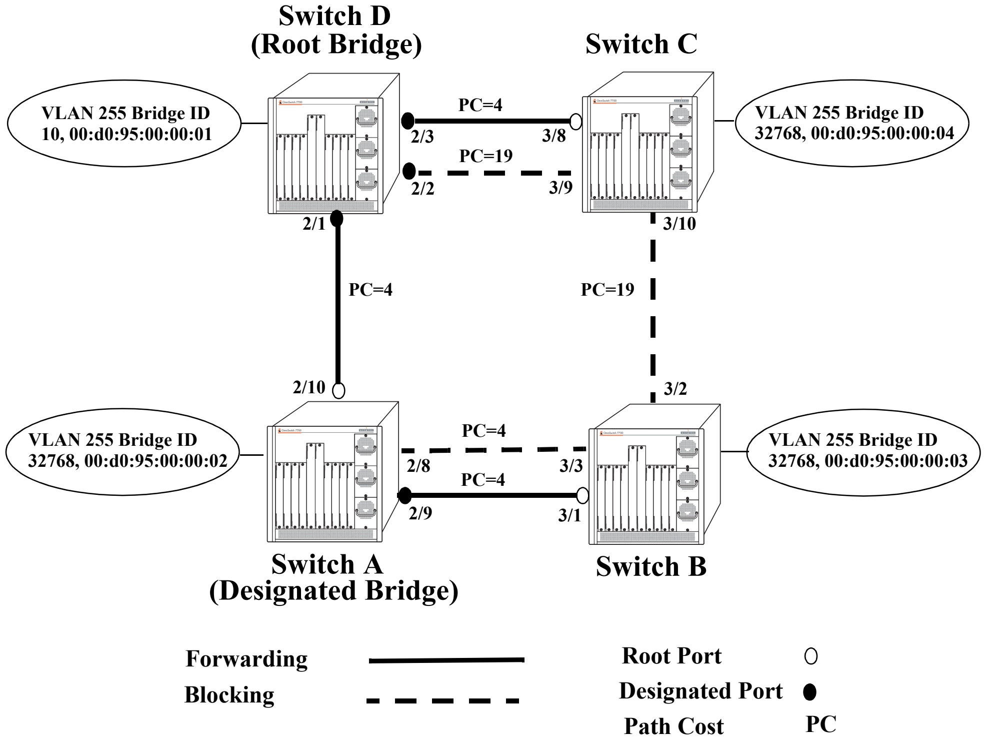
Sample Spanning Tree Configuration

This section provides an example network configuration in which Spanning Tree has calculated a loop-free topology. In addition, a tutorial is also included that provides steps on how to configure the example network topology using the Command Line Interface (CLI).

Note that the following example network configuration illustrates using switches operating in the 1x1 Spanning Tree mode and using RSTP (802.1w) to calculate a single data path between VLANs. See Chapter 3, "Using 802.1s Multiple Spanning Tree," for an overview and examples of using MSTP (802.1s).

Example Network Overview

The following diagram shows a four-switch network configuration with an active Spanning Tree topology, which was calculated based on both configured and default Spanning Tree parameter values:



Example Active Spanning Tree Topology

In the above example topology:

- Each switch is operating in the 1x1 Spanning Tree mode by default.
- Each switch configuration has a VLAN 255 defined. The Spanning Tree administrative status for this VLAN was enabled by default when the VLAN was created.
- VLAN 255 on each switch is configured to use the 802.1w (rapid reconfiguration) Spanning Tree Algorithm and Protocol.

- Ports 2/1-3, 2/8-10, 3/1-3, and 3/8-10 provide connections to other switches and are all assigned to VLAN 255 on their respective switches. The Spanning Tree administrative status for each port is enabled by default.
- The path cost for each port connection defaults to a value based on the link speed. For example, the connection between Switch B and Switch C is a 100 Mbps link, which defaults to a path cost of 19.
- VLAN 255 on Switch D is configured with a Bridge ID priority value of 10, which is less than the same value for VLAN 255 configured on the other switches. As a result, VLAN 255 was elected the Spanning Tree root bridge for the VLAN 255 broadcast domain.
- A root port is identified for VLAN 255 on each switch, except the root VLAN 255 switch. The root port identifies the port that provides the best path to the root VLAN.
- VLAN 255 on Switch A was elected the designated bridge because it offers the best path cost for Switch B to the root VLAN 255 on Switch D.
- Port 2/9 on Switch A is the designated port for the Switch A to Switch B connection because Switch A is the designated bridge for Switch B.
- Redundant connections exist between Switch D and Switch C. Ports 2/2 and 3/9 are in a discarding (blocking) state because this connection has a higher path cost than the connection provided through ports 2/3 and 3/8. As a result, a network loop condition is avoided.
- Redundant connections also exist between Switch A and Switch B. Although the path cost value for both of these connections is the same, ports 2/8 and 3/3 are in a discarding state because their port priority values (not shown) are higher than the same values for ports 2/10 and 3/1.
- The ports that provide the connection between Switch B and Switch C are in a discarding (blocking) state, because this connection has a higher path cost than the other connections leading to the root VLAN 255 on Switch D. As a result, a network loop is avoided.

Example Network Configuration Steps

The following steps provide a quick tutorial that configures the active Spanning Tree network topology shown in the diagram on page 6-30.

1 Create VLAN 255 on Switches A, B, C, and D with “Marketing IP Network” for the VLAN description on each switch using the following command:

```
-> vlan 255 name "Marketing IP Network"
```

2 Assign the switch ports that provide connections between each switch to VLAN 255. For example, the following commands entered on Switches A, B, C, and D, respectively, assign the ports shown in the example network diagram on page 6-30 to VLAN 255:

```
-> vlan 255 port default 2/8-10
-> vlan 255 port default 3/1-3
-> vlan 255 port default 3/8-10
-> vlan 255 port default 2/1-3
```

3 Change the Spanning Tree protocol for VLAN 255 to 802.1w (rapid reconfiguration) on each switch using the following command:

```
-> bridge 255 protocol 1w
```

4 Change the bridge priority value for VLAN 255 on Switch D to **10** using the following command (leave the priority for VLAN 255 on the other three switches set to the default value of **32768**):

```
-> bridge 255 priority 10
```

VLAN 255 on Switch D will have the lowest Bridge ID priority value of all four switches, which will qualify it as the Spanning Tree root VLAN for the VLAN 255 broadcast domain.

Note. To verify the VLAN 255 Spanning Tree configuration on each switch use the following show commands. The following outputs are for example purposes only and may not match values shown in the sample network configuration:

```
-> show spantree 255
Spanning Tree Parameters for Vlan 255
Spanning Tree Status :                   ON,
Protocol             :          IEEE 802.1W (Fast STP),
mode                 : 1X1 (1 STP per Vlan),
Priority             :        32768(0x0FA0),
Bridge ID            :   8000-00:d0:95:00:00:04,
Designated Root      :   000A-00:d0:95:00:00:01,
Cost to Root Bridge  :                    4,
Root Port            :   Slot 3 Interface 8,
Next Best Root Cost  :                    0,
Next Best Root Port  :                 None,
Hold Time            :                    1,
Topology Changes     :                    3,
Topology age         :          0:4:37
Current Parameters (seconds)
Max Age              =    30,
Forward Delay        =    15,
Hello Time           =     2
Parameters system uses when attempting to become root
System Max Age       =    30,
System Forward Delay =    15,
System Hello Time    =       2

-> show spantree 255 ports
Spanning Tree Port Summary for Vlan 255
      Oper   Path  Desig         Fw   Prim. Op
Port   St    Cost  Cost   Role   Tx   Port  Cnx  Desig Bridge ID
-----+-----+------+------+-----+-----+-----+----+-------------------
 3/8  FORW      4     29  ROOT   1     3/8   NPT 000A-00:d0:95:00:00:01
 3/9  BLOCK    19     48  BACK   0     3/9   NPT 8000-00:d0:95:00:00:04
 3/10 BLOCK    19     48  ALTN   0     3/10  NPT 8000-00:d0:95:00:00:03
```

Verifying the Spanning Tree Configuration

To display information about the Spanning Tree configuration on the switch, use the show commands listed below:

show spantree	Displays VLAN Spanning Tree information, including parameter values and topology change statistics.
show spantree ports	Displays Spanning Tree information for switch ports, including parameter values and the current port state.

For more information about the resulting displays from these commands, see the *OmniSwitch CLI Reference Guide*. An example of the output for the **show spantree** and **show spantree ports** commands is also given in "Example Network Configuration Steps" on page 6-31.

7 Assigning Ports to VLANs

Initially all switch ports are non-mobile and are assigned to VLAN 1, which is also their *configured default* VLAN. When additional VLANs are created on the switch, ports are assigned to the VLANs so that traffic from devices connected to these ports is bridged within the VLAN domain. Switch ports are either statically or dynamically assigned to VLANs.

Methods for statically assigning ports to VLANs include the following:

- Using the **vlan port default** command to define a new configured default VLAN for both non-mobile (fixed) and mobile ports. (See "Statically Assigning Ports to VLANs" on page 7-4.)
- Using the **vlan 802.1q** command to define tagged VLANs for non-mobile ports. This method allows the switch to bridge traffic for multiple VLANs over one physical port connection. (See Chapter 10, "Configuring 802.1Q.")
- Configuring ports as members of a link aggregate that is assigned to a configured default VLAN. (See Chapter 11, "Configuring Static Link Aggregation," and Chapter 12, "Configuring Dynamic Link Aggregation.")

Dynamic assignment applies only to mobile ports and requires the additional configuration of VLAN rules. When traffic is received on a mobile port, the packets are examined to determine if their content matches any VLAN rules configured on the switch. If a match occurs, the mobile port is automatically assigned to the VLAN without user intervention. (See "Dynamically Assigning Ports to VLANs" on page 7-4 for more information.)

Regardless of how a port is assigned to a VLAN, once the assignment occurs, a VLAN port association (VPA) is created and tracked by VLAN management software on each switch.

In This Chapter

This chapter describes how to statically assign ports to a new default VLAN and configure mobile ports for dynamic assignment through the Command Line Interface (CLI). CLI commands are used in the configuration examples; for more details about the syntax of commands, see the *OmniSwitch CLI Reference Guide*.

Configuration procedures described in this chapter include:

- Statically assigning ports to VLANs on page 7-4.
- Configuring port mobility on page 7-8.
- Configuring mobile port properties (including authentication) on page 7-13.

Port Assignment Specifications

IEEE Standards Supported	802.1Q–*Virtual Bridged Local Area Networks* 802.1D–*Media Access Control Bridges*
Maximum VLANs per switch	4094 (including default VLAN 1)
Maximum VLAN port associations	32768
Switch ports eligible for port mobility.	Untagged 10/100 Ethernet and gigabit ports that are not members of a link aggregate.
Switch ports eligible for dynamic VLAN assignment.	Mobile ports.
Switch ports eligible for static VLAN assignment.	Non-mobile (fixed) ports. Mobile ports. Link aggregate of ports.

Port Assignment Defaults

Parameter Description	Command	Default
Configured default VLAN	**vlan port default**	All ports initially associated with default VLAN 1.
Port mobility	**vlan port mobile**	Disabled
Bridge mobile port traffic that doesn't match any VLAN rules on the configured default VLAN	**vlan port default vlan**	Enabled
Drop mobile port dynamic VLAN assignments when learned mobile port traffic that triggered the assignment ages out	**vlan port default vlan restore**	Enabled
Enable Layer 2 authentication on a mobile port	**vlan port authenticate**	Disabled
Enable 802.1x port-based access control on a mobile port	**vlan port 802.1x**	Disabled

Sample VLAN Port Assignment

The following steps provide a quick tutorial that will create a VLAN, statically assign ports to the VLAN, and configure mobility on some of the VLAN ports:

1 Create VLAN 255 with a description (e.g., Finance IP Network) using the following command:

```
-> vlan 255 name "Finance IP Network"
```

2 Assign switch ports 2 through 5 on slot 3 to VLAN 255 using the following command:

```
-> vlan 255 port default 3/2-5
```

VLAN 255 is now the *configured default VLAN* for ports 2 through 5 on slot 3.

3 Enable mobility on ports 4 and 5 on slot 3 using the following command:

```
-> vlan port mobile 3/4-5
```

4 Disable the default VLAN parameter for mobile ports 3/4 and 3/5 using the following command:

```
-> vlan port 3/4-5 default vlan disable
```

With this parameter disabled, VLAN 255 will not carry any traffic received on 3/4 or 3/5 that does not match any VLAN rules configured on the switch.

Note. *Optional*. To verify that ports 2 through 5 on slot 3 were assigned to VLAN 255, enter **show vlan** followed by 255 then **port**. For example:

```
-> show vlan 255 port
  port     type      status
--------+---------+--------------
   3/2    default   inactive
   3/3    default   inactive
   3/4    default   inactive
   3/5    default   inactive
```

To verify the mobile status of ports 4 and 5 on slot 3 and determine which mobile port parameters are enabled, enter **show vlan port mobile** followed by a slot and port number. For example:

```
-> show vlan port mobile 3/4
Mobility            : on,
Config  Default Vlan: 255,
Default Vlan Enabled: off,
Default Vlan Perm   : on,
Default Vlan Restore: on,
Authentication      : off,
Ignore BPDUs        : off
```

Statically Assigning Ports to VLANs

The **vlan port default** command is used to statically assign both mobile and non-mobile ports to another VLAN. When the assignment is made, the port drops the previous VLAN assignment. For example, the following command assigns port 2 on slot 3, currently assigned to VLAN 1, to VLAN 755:

```
-> vlan 755 port default 3/2
```

Port 3/2 is now assigned to VLAN 755 and no longer associated with VLAN 1. In addition, VLAN 755 is now the new configured default VLAN for the port.

A configured default VLAN is the VLAN statically assigned to a port. Any time the **vlan port default** command is used, the VLAN assignment is static and a new configured default VLAN is defined for the port. This command is also the only way to change a non-mobile port VLAN assignment. In addition, non-mobile ports can only retain one VLAN assignment, unlike mobile ports that can dynamically associate with multiple VLANs. See "Dynamically Assigning Ports to VLANs" on page 7-4 for more information about mobile ports.

Additional methods for statically assigning ports to VLANs include the following:

- Using the **vlan 802.1q** command to define tagged VLANs for non-mobile ports. This method allows the switch to bridge traffic for multiple VLANs over one physical port connection. (See Chapter 10, "Configuring 802.1Q," for more information.)
- Configuring ports as members of a link aggregate that is assigned to a configured default VLAN. (See Chapter 11, "Configuring Static Link Aggregation," and Chapter 12, "Configuring Dynamic Link Aggregation," for more information.)

When a port is statically assigned to a VLAN, a VLAN port association (VPA) is created and tracked by VLAN management software on each switch. To display a list of all VPAs, use the **show vlan port** command. For more information, see "Verifying VLAN Port Associations and Mobile Port Properties" on page 7-16.

Dynamically Assigning Ports to VLANs

Mobile ports are the only types of ports that are eligible for dynamic VLAN assignment. When traffic received on a mobile port matches pre-defined VLAN rules, the port and the matching traffic are assigned to the VLAN without user intervention.

By default, all switch ports are non-mobile (fixed) ports that are statically assigned to a specific VLAN and can only belong to one default VLAN at a time. The **vlan port mobile** command is used to enable mobility on a port. Once enabled, switch software classifies mobile port traffic to determine the appropriate VLAN assignment.

VLANs do not have a mobile or non-mobile distinction and there is no overall switch setting to invoke the mobile port feature. Instead, mobility is enabled on individual switch ports and rules are defined for individual VLANs to classify mobile port traffic.

When a port is dynamically assigned to a VLAN, a VLAN port association (VPA) is created and tracked by VLAN management software on each switch. To display a list of all VPAs, use the **show vlan port** command. For more information, see "Verifying VLAN Port Associations and Mobile Port Properties" on page 7-16.

VLAN Rule Classification

VLAN rule classification triggers dynamic VLAN port assignment when traffic received on a mobile port matches the criteria defined in a VLAN rule. Different rule types are available for classifying different types of network device traffic (see Chapter 8, "Defining VLAN Rules," for more information).

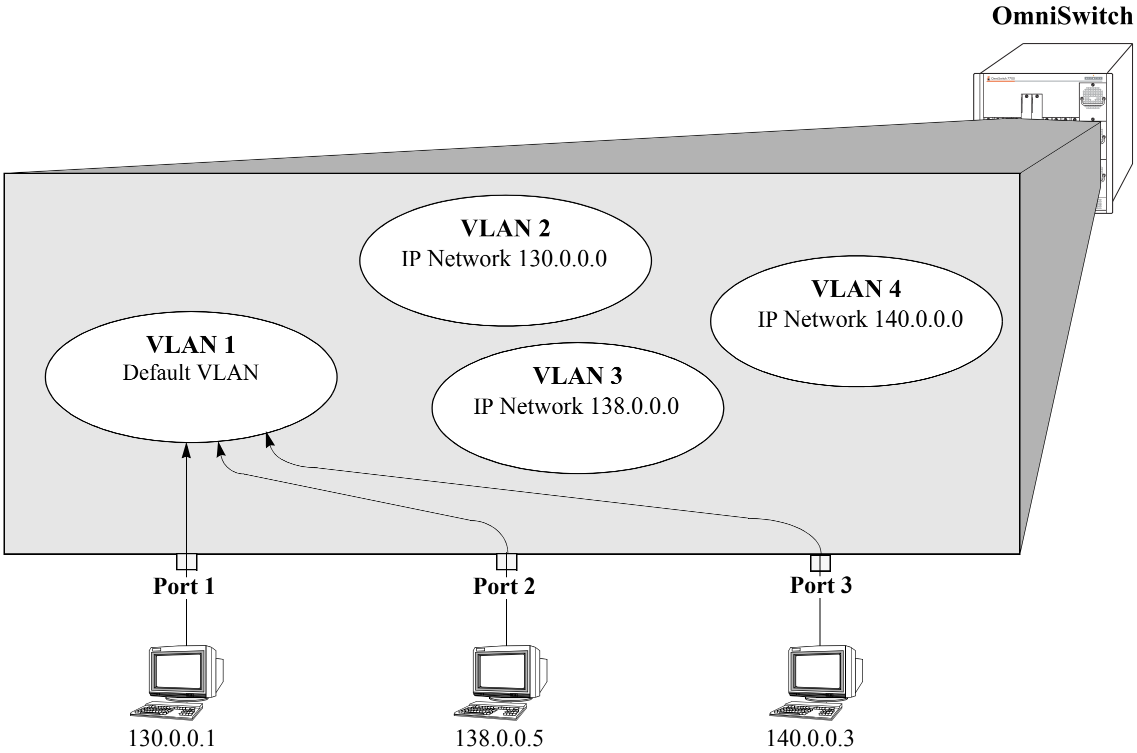
Consider the following when configuring rules for a VLAN:

- The VLAN must already exist. Use the **vlan** command to create a new VLAN or the **show vlan** command to verify a VLAN is already configured. Refer to Chapter 5, "Configuring VLANs," for more information.
- Which type of rule to define; DHCP, binding, MAC address, protocol, network address, custom, or port.
- What is the rule's precedence compared to other rules defined for other VLANs. If mobile port traffic matches rules defined for more than one VLAN, the mobile port is dynamically assigned to the VLAN with the higher precedence rule.
- It is possible to define multiple rules of different types for one VLAN and rules for multiple VLANs.
- It is possible for mobile ports to belong to more than one VLAN when the port carries multiple traffic types that match different rules on different VLANs.
- There is no limit to the number of rules defined for a single VLAN and up to 8,129 of each rule type is allowed per switch.
- It is possible to create a custom rule or protocol rules based on Ether type, SNAP type, or DSAP/SSAP values. It is recommended, however, to use predefined rules (such as MAC address, network address, and generic protocol rules) whenever possible to ensure accurate results when capturing mobile port traffic.
- When a VLAN is administratively disabled, static port and dynamic mobile port assignments are retained but traffic on these ports is not forwarded. However, VLAN rules remain active and continue to classify mobile port traffic for VLAN membership.
- When a VLAN is deleted from the switch configuration, all rules defined for that VLAN are automatically removed and any static or dynamic port assignments are dropped.

The following example illustrates how mobile ports are dynamically assigned using VLAN rules to classify mobile port traffic. This example includes diagrams showing the initial VLAN port assignment configuration and a diagram showing how the configuration looks after mobile port traffic is classified.

In the initial VLAN port assignment configuration shown on page 7-6,

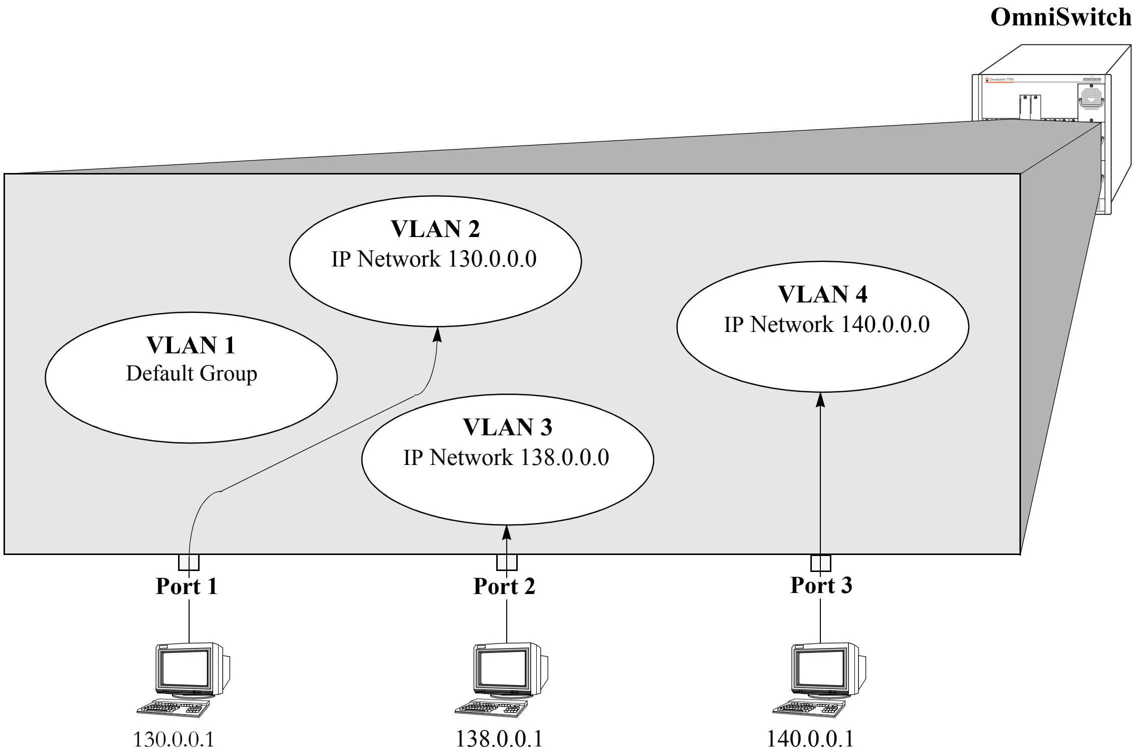
- All three ports have workstations that belong to three different IP subnets (130.0.0.0, 138.0.0.0, and 140.0.0.0).
- Mobility is enabled on each of the workstation ports.
- VLAN 1 is the configured default VLAN for each port.
- Three additional VLANs are configured on the switch, each one has an IP network address rule defined for one of the IP subnets.



Initial Configuration: All Ports in Default VLAN

As soon as the workstations start sending traffic, switch software checks the source subnet of the frames and looks for a match with any configured IP network address rules. Since the workstations are sending traffic that matches a VLAN rule, each port is assigned to the appropriate VLAN without user intervention. As the following diagram on page 7-7 shows.

- Port 1 is assigned to VLAN 2, because the workstation is transmitting IP traffic on network 130.0.0.0 that matches the VLAN 2 network address rule.
- Port 2 is assigned to VLAN 3 because the workstation is transmitting IP traffic on network 138.0.0.0 that matches the VLAN 3 network address rule.
- Port 3 is assigned to VLAN 4 because the workstation is transmitting IP traffic on network 140.0.0.0 that matches the VLAN 4 network address rule.



Mobile Port Traffic Triggers Dynamic VLAN Assignment

Configuring Dynamic VLAN Port Assignment

Dynamic VLAN port assignment requires the following configuration steps:

1 Use the **vlan port mobile** command to enable mobility on switch ports that will participate in dynamic VLAN assignment. See "Enabling/Disabling Port Mobility" on page 7-8 for detailed procedures.

2 Enable/disable mobile port properties that determine mobile port behavior. See "Configuring Mobile Port Properties" on page 7-13 for detailed procedures.

3 Create VLANs that will receive and forward mobile port traffic. See Chapter 5, "Configuring VLANs," for more information.

4 Define VLAN rules for the VLANs that were created in Step 3. These rules will trigger dynamic assignment of mobile ports to these VLANs. See "VLAN Rule Classification" on page 7-5.

Once the above configuration steps are completed, dynamic VLAN assignment occurs when a device connected to a mobile port starts to send traffic. This traffic is examined by switch software to determine which VLAN should carry the traffic based on VLAN rule criteria.

See "Dynamically Assigning Ports to VLANs" on page 7-4 for more information and examples of dynamic VLAN port assignment.

Enabling/Disabling Port Mobility

To enable the mobility feature on a port, use the **vlan port mobile** command. For example, the following command enables mobility on port 1 of slot 4:

```
-> vlan port mobile 4/1
```

To enable mobility on multiple ports, specify a range of ports and/or multiple slots.

```
-> vlan port mobile 4/1-5 5/12-20 6/10-15
```

Use the **no** form of this command to disable port mobility.

```
-> vlan no port mobile 5/21-24 6/1-4
```

Only 10/100 Ethernet and gigabit Ethernet ports are eligible to become mobile ports. If any of the following conditions are true, however, these ports are considered non-mobile ports and are not available for dynamic VLAN assignment:

- The mobile status for the port is disabled (the default).
- The port is an 802.1Q tagged port.
- The port belongs to a link aggregate of ports.
- Spanning Tree is active on the port and the BPDU ignore status is disabled for the port. (See "Ignoring Bridge Protocol Data Units (BPDU)" on page 7-8 for more information.)
- The port is configured to mirror other ports.

Note. Mobile ports are automatically *trusted* ports regardless of the QoS settings. See Chapter 24, "Configuring QoS," for more information.

Use the **show vlan port mobile** command to display a list of ports that are mobile or are eligible to become mobile. For more information about this command, see the *OmniSwitch CLI Reference Guide*.

Ignoring Bridge Protocol Data Units (BPDU)

By default, ports that send or receive spanning tree Bridge Protocol Data Units (BPDU) are not eligible for dynamic VLAN assignment. If the switch sees BPDU on a port, it does not attempt to classify the port's traffic by comparing the traffic to existing VLAN rules. The **vlan port mobile** command, however, provides an optional **BPDU ignore** parameter. If this parameter is enabled when mobility is enabled on the port, the switch does not look for BPDU to determine if the port is eligible for dynamic assignment.

When **BPDU ignore** is disabled and the mobile port receives a BPDU, mobility is shut off on the port and the following occurs:

- The Switch Logging feature is notified of the port's change in mobile status (see Chapter 30, "Using Switch Logging," for more information).
- The port becomes a fixed (non-mobile) port that is associated only with its configured default VLAN.
- The port is included in the Spanning Tree algorithm.
- Mobility remains off on the port even if the port's link is disabled or disconnected. Rebooting the switch, however, will restore the port's original mobile status.

When **BPDU ignore** is enabled and the mobile port receives a BPDU, the following occurs:

- The port retains its mobile status and remains eligible for dynamic VLAN assignment.
- The port is not included in the Spanning Tree algorithm.

Note. Enabling BPDU ignore is not recommended. In specific cases where it is required, such as connecting legacy networks to mobile port networks, make sure that ignoring BPDU on a mobile port will not cause network loops to go undetected. Connectivity problems could also result if a mobile BPDU port dynamically moves out of its configured default VLAN where it provides traffic flow to/from the network.

The following command enables mobility and BPDU ignore on port 8 of slot 3:

```
-> vlan port mobile 3/8 BPDU ignore enable
```

Enabling mobility on an active port that sends or receives BPDU (e.g. ports that connect two switches and Spanning Tree is enabled on both the ports and their assigned VLANs) is not allowed. If mobility is required on this type of port, enable mobility and the **BPDU ignore** parameter when the port is not active.

Understanding Mobile Port Properties

Dynamic assignment of mobile ports occurs without user intervention when mobile port traffic matches VLAN rule criteria. When ports are dynamically assigned, however, the following configurable mobile port properties affect how a port uses its *configured default VLAN* and how long it retains a VLAN port association (VPA):

Mobile Port Property	If enabled	If disabled
Default VLAN	Port traffic that does not match any VLAN rules configured on the switch is flooded on the port's configured default VLAN.	Port traffic that does not match any VLAN rules is discarded.
Restore default VLAN	Port does not retain a dynamic VPA when the traffic that triggered the assignment ages out of the switch MAC address table (forwarding database).	Port retains a dynamic VPA when the qualifying traffic ages out of the switch MAC address table.

The effects of enabling or disabling mobile port properties are described through the following diagrams:

- How Mobile Port Traffic that Does Not Match any VLAN Rules is Classified on page 7-11.
- How Mobile Port VLAN Assignments Age on page 7-12.

What is a Configured Default VLAN?

Every switch port, mobile or non-mobile, has a configured default VLAN. Initially, this is VLAN 1 for all ports, but is configurable using the **vlan port default** command. For more information, see "Statically Assigning Ports to VLANs" on page 7-4.

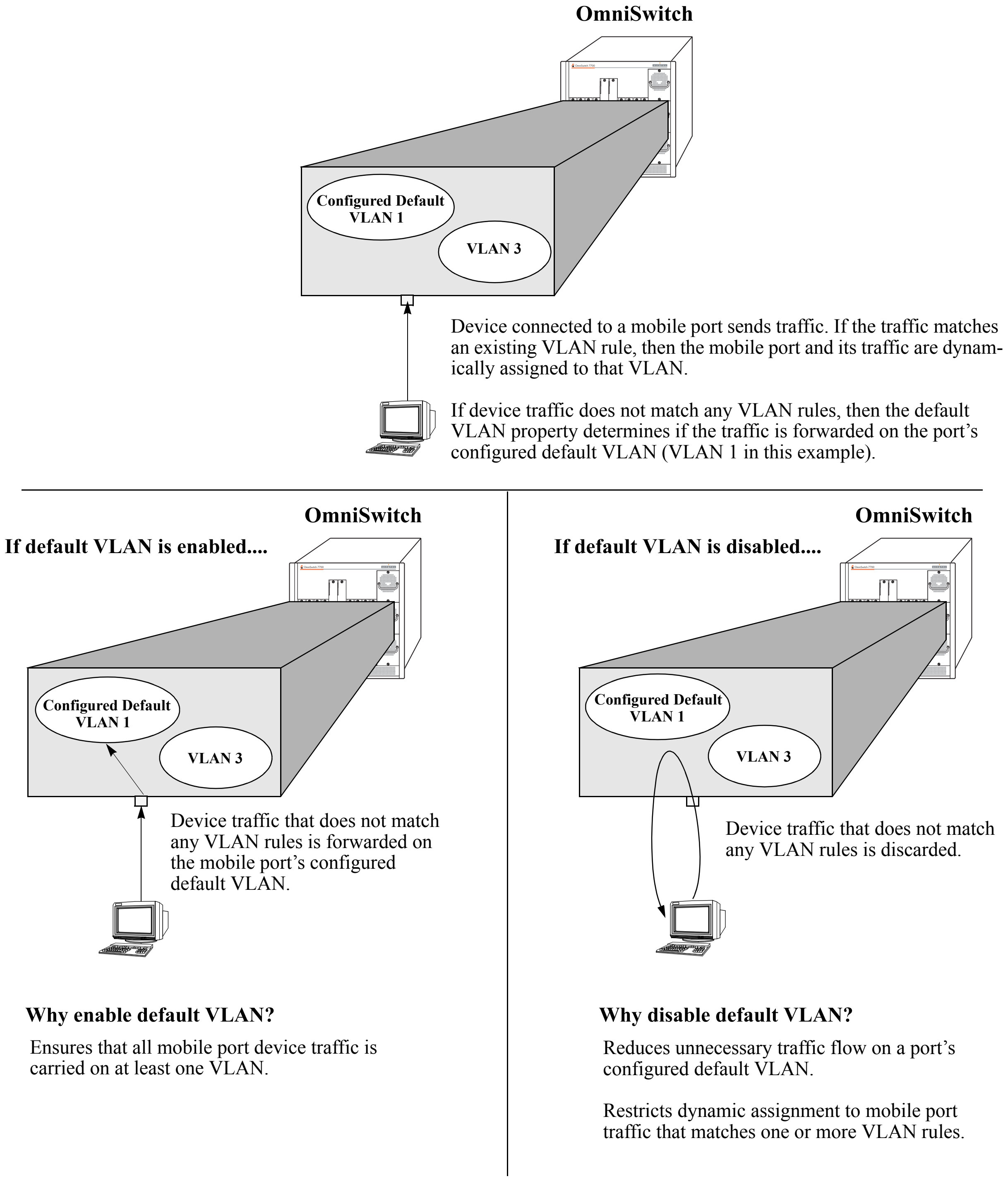
To view current VPA information for the switch, use the **show vlan port** command. Configured default VLAN associations are identified with a value of **default** in the **type** field. For more information, see "Verifying VLAN Port Associations and Mobile Port Properties" on page 7-16.

What is a Secondary VLAN?

One of the main benefits of dynamic port assignment is that mobile ports can join more than one VLAN. For example, mobile ports start out with a configured default VLAN assignment. When port traffic matches a VLAN rule, the port is assigned to that VLAN. If traffic matches rules on more than one VLAN, then the port joins each of those as well. Secondary VLANs are any VLAN a port is subsequently assigned to that is not the configured default VLAN for that port.

VLAN Management software on each switch tracks VPAs. When a mobile port link is disabled and then enabled, all secondary VLAN assignments for that port are automatically dropped. Switch ports are disabled when a device is disconnected from the port, a configuration change is made to disable the port, or switch power is turned off.

To view current VPA information for the switch, use the **show vlan port** command. Dynamic secondary VLAN associations are identified with a value of **mobile** in the **type** field. For more information, see "Verifying VLAN Port Associations and Mobile Port Properties" on page 7-16.


OmniSwitch
Configured Default VLAN 1
VLAN 3
OmniSwitch
If default VLAN is enabled....
Configured Default VLAN 1
VLAN 3
OmniSwitch
If default VLAN is disabled....
Configured Default VLAN 1
VLAN 3

Device connected to a mobile port sends traffic. If the traffic matches an existing VLAN rule, then the mobile port and its traffic are dynamically assigned to that VLAN.

If device traffic does not match any VLAN rules, then the default VLAN property determines if the traffic is forwarded on the port's configured default VLAN (VLAN 1 in this example).

Device traffic that does not match any VLAN rules is forwarded on the mobile port's configured default VLAN.

Device traffic that does not match any VLAN rules is discarded.

Why enable default VLAN?

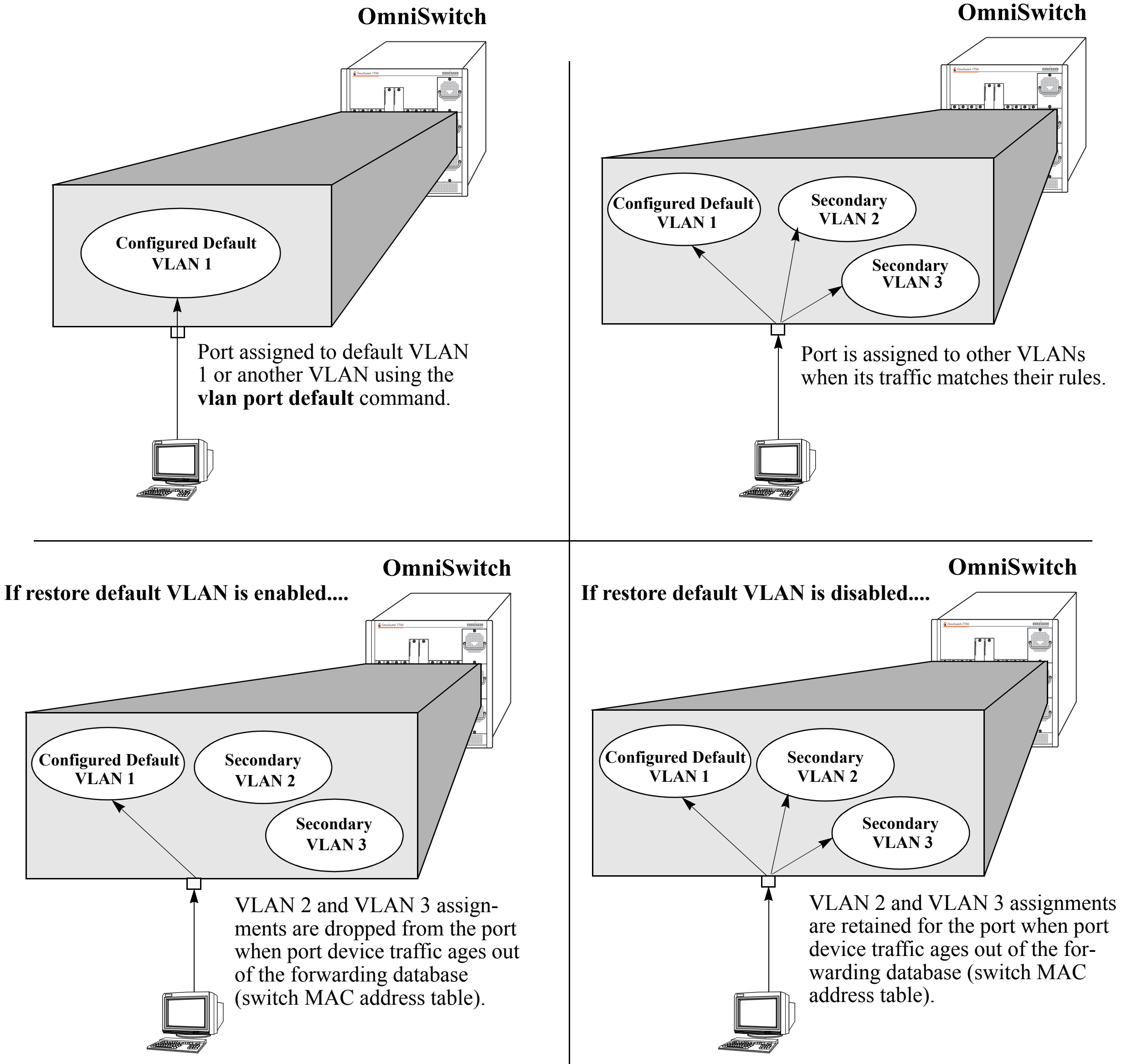
Ensures that all mobile port device traffic is carried on at least one VLAN.

Why disable default VLAN?

Reduces unnecessary traffic flow on a port's configured default VLAN.

Restricts dynamic assignment to mobile port traffic that matches one or more VLAN rules.

How Mobile Port Traffic that Does Not Match any VLAN Rules is Classified


OmniSwitch
Configured Default VLAN 1
OmniSwitch
Configured Default VLAN 1
Secondary VLAN 2
Secondary VLAN 3
OmniSwitch
If restore default VLAN is enabled....
Configured Default VLAN 1
Secondary VLAN 2
Secondary VLAN 3
OmniSwitch
If restore default VLAN is disabled....
Configured Default VLAN 1
Secondary VLAN 2
Secondary VLAN 3

Port assigned to default VLAN 1 or another VLAN using the **vlan port default** command.

Port is assigned to other VLANs when its traffic matches their rules.

VLAN 2 and VLAN 3 assignments are dropped from the port when port device traffic ages out of the forwarding database (switch MAC address table).

VLAN 2 and VLAN 3 assignments are retained for the port when port device traffic ages out of the forwarding database (switch MAC address table).

Why enable restore default VLAN?

Security. VLANs only contain mobile port traffic that has recently matched rule criteria.

VPAs created from occasional network users (e.g., laptop) are not unnecessarily retained.

Why disable restore default VLAN?

VPAs are retained even when port traffic is idle for some time. When traffic resumes, it is not necessary to relearn the same VPA again. Appropriate for devices that only send occasional traffic.

How Mobile Port VLAN Assignments Age

Configuring Mobile Port Properties

Mobile port properties indicate mobile port status and affect port behavior when the port is dynamically assigned to one or more VLANs. For example, mobile port properties determine the following:

- Should the configured default VLAN forward or discard port traffic that does not match any VLAN rule criteria.
- Should the port retain or drop a dynamic VPA when traffic that triggered the assignment stops and the source MAC address learned on the port for that VLAN is aged out. (See Chapter 2, "Managing Source Learning," for more information about the aging of MAC addresses.)
- Will the mobile port participate in Layer 2 authentication that provides a login process at the VLAN and/or port level. (See Chapter 21, "Configuring Authenticated VLANs," and Chapter 22, "Configuring 802.1X," for more information.)

This section contains procedures for using the following commands to configure mobile port properties. For more details about the syntax of commands, see the *OmniSwitch CLI Reference Guide*.

Command	Description
vlan port default vlan	Enables or disables forwarding of mobile port traffic on the port's configured default VLAN that does not match any existing VLAN rules.
vlan port default vlan restore	Enables or disables the retention of VLAN port assignments when mobile port traffic ages out.
vlan port authenticate	Enables or disables authentication on a mobile port.
vlan port 802.1x	Enables or disables 802.1X port-based access control on a mobile port.

Use the **show vlan port mobile** command to view the current status of these properties for one or more mobile ports. See "Verifying VLAN Port Associations and Mobile Port Properties" on page 7-16 for more information.

Enable/Disable Default VLAN

To enable or disable forwarding of mobile port traffic that does not match any VLAN rules on the port's configured default VLAN, enter **vlan port** followed by the port's **slot/port** designation then **default vlan** followed by **enable** or **disable**. For example,

```
-> vlan port 3/1 default vlan enable
-> vlan port 5/2 default vlan disable
```

To enable or disable the configured default VLAN on multiple ports, specify a range of ports and/or multiple slots.

```
-> vlan port 6/1-32 8/10-24 9/3-14 default vlan enable
```

Note. It is recommended that mobile ports with their default VLAN disabled should not share a VLAN with any other types of ports (e.g., mobile ports with default VLAN enabled or non-mobile, fixed ports).

See "Understanding Mobile Port Properties" on page 7-10 for an overview and illustrations of how this property affects mobile port behavior.

Enable/Disable Default VLAN Restore

To enable or disable default VLAN restore, enter **vlan port** followed by the port's **slot/port** designation then **default vlan restore** followed by **enable** or **disable**. For example,

```
-> vlan port 3/1 default vlan restore enable
-> vlan port 5/2 default vlan restore disable
```

To enable or disable default VLAN restore on multiple ports, specify a range of ports and/or multiple slots.

```
-> vlan port 6/1-32 8/10-24 9/3-14 default vlan restore enable
```

Note the following when changing the restore default VLAN status for a mobile port:

- If a hub is connected to a mobile port, enabling default VLAN restore on that port is recommended.
- VLAN port rule assignments are exempt from the effects of the restore default VLAN status. See Chapter 8, "Defining VLAN Rules," for more information about using port rules to forward mobile port traffic
- When a mobile port link is disabled and then enabled, all secondary VPAs for that port are dropped regardless of the restore default VLAN status for that port. Switch ports are disabled when a device is disconnected from the port, a configuration change is made to disable the port, or switch power is turned off.

See "Understanding Mobile Port Properties" on page 7-10 for an overview and illustrations of how this property affects mobile port behavior.

Enable/Disable Port Authentication

To enable or disable authentication on a mobile port, enter **vlan port** followed by the port's **slot/port** designation then **authenticate** followed by enable or disable. For example,

```
-> vlan port 3/1 authenticate enable
-> vlan port 5/2 authenticate disable
```

To enable or disable authentication on multiple ports, specify a range of ports and/or multiple slots.

```
-> vlan port 6/1-32 8/10-24 9/3-14 authenticate enable
-> vlan port 5/3-6 9/1-4 authenticate disable
```

Only mobile ports are eligible for authentication. If enabled, the mobile port participates in the Layer 2 authentication process supported by Alcatel switches. This process restricts switch access at the VLAN level. The user is required to enter a valid login ID and password before gaining membership to a VLAN. For more information, see Chapter 21, "Configuring Authenticated VLANs."

Enable/Disable 802.1X Port-Based Access Control

To enable or disable 802.1X on a mobile port, enter **vlan port** followed by the port's **slot/port** designation then **802.1x** followed by enable or disable. For example,

```
-> vlan port 3/1 802.1x enable
-> vlan port 5/2 802.1x disable
```

To enable or disable 802.1X on multiple ports, specify a range of ports and/or multiple slots.

```
-> vlan port 6/1-32 8/10-24 9/3-14 802.1x enable
-> vlan port 5/3-6 9/1-4 802.1x disable
```

Only mobile ports are eligible for 802.1X port-based access control. If enabled, the mobile port participates in the authentication and authorization process defined in the IEEE 802.1X standard and supported by Alcatel switches. For more information, see Chapter 22, "Configuring 802.1X."

Verifying VLAN Port Associations and Mobile Port Properties

To display a list of VLAN port assignments or the status of mobile port properties, use the show commands listed below:

show vlan port	Displays a list of VLAN port assignments, including the type and status for each assignment
show vlan port mobile	Displays the mobile status and current mobile parameter values for each port.

Understanding 'show vlan port' Output

Each line of the **show vlan port** command display corresponds to a single VLAN port association (VPA). In addition to showing the VLAN ID and slot/port number, the VPA type and current status of each association are also provided.

The VPA type indicates that one of the following methods was used to create the VPA:

Type	Description
default	The port was statically assigned to the VLAN using the **vlan port default** command. The VLAN is now the port's configured default VLAN.
qtagged	The port was statically assigned to the VLAN using the **vlan 802.1q** command. The VLAN is a static secondary VLAN for the 802.1Q tagged port.
mobile	The port is mobile and was dynamically assigned when traffic received on the port matched traffic rules defined for the VLAN. The VLAN is a dynamic secondary VLAN assignment for the mobile port.
mirror	The port is assigned to the VLAN because it is configured to mirror another port that is assigned to the same VLAN. For more information about the Port Mirroring feature, see Chapter 29, "Diagnosing Switch Problems."

The VPA status indicates one of the following:

Status	Description
inactive	Port is not active (administratively disabled, down, or nothing connected to the port) for the VPA.
blocking	Port is active, but not forwarding traffic for the VPA.
forwarding	Port is forwarding all traffic for the VPA.
filtering	Mobile port traffic is filtered for the VPA; only traffic received on the port that matches VLAN rules is forwarded. Occurs when a mobile port's VLAN is administratively disabled or the port's default VLAN status is disabled. Does not apply to fixed ports.

The following example uses the **show vlan port** command to display VPA information for all ports in VLAN 200:

```
-> show vlan 200 port

  port     type      status
--------+---------+--------------
   3/24   default    inactive
   5/11   mobile     forwarding
   5/12   qtagged    blocking
```

The above example output provides the following information:

- VLAN 200 is the configured default VLAN for port 3/24, which is currently not active.
- VLAN 200 is a secondary VLAN for mobile port 5/11, which is currently forwarding traffic for this VPA.
- VLAN 200 is an 802.1Q tagged VLAN for port 5/12, which is an active port but currently blocked from forwarding traffic.

Another example of the output for the **show vlan port** command is also given in "Sample VLAN Port Assignment" on page 7-3. For more information about the resulting display from this command, see the *OmniSwitch CLI Reference Guide*.

Understanding 'show vlan port mobile' Output

The **show vlan port mobile** command provides information regarding a port's mobile status. If the port is mobile, the resulting display also provides the current status of the port's mobile properties. The following example displays mobile port status and property values for ports 8/2 through 8/5:

```
-> show vlan port mobile
                 cfg                               ignore
  port   mobile  def  authent   enabled   restore   bpdu
-------+--------+----+--------+---------+---------+-------
  8/2      on     200    off      off         on     off
  8/3      on     200    off       on        off     off
  8/4      on     200 on-avlan    off         on     off
  8/5      on     200 on-8021x     on        off     off
```

Note that the **show vlan port mobile** command only displays ports that are mobile or are eligible to become mobile ports. For example, ports that are part of a link aggregate or are configured for 802.1Q VLAN tagging are not included in the output of this command.

Another example of the output for the **show vlan port mobile** command is also given in "Sample VLAN Port Assignment" on page 7-3. For more information about the resulting display from this command, see the *OmniSwitch CLI Reference Guide*.

8 Defining VLAN Rules

VLAN rules are used to classify mobile port traffic for dynamic VLAN port assignment. Rules are defined by specifying a port, MAC address, protocol, network address, user-defined, binding, or DHCP criteria to capture certain types of network device traffic. It is also possible to define multiple rules for the same VLAN. A mobile port is assigned to a VLAN if its traffic matches any one VLAN rule.

In This Chapter

This chapter describes how to define VLAN rules through the Command Line Interface (CLI). CLI commands are used in the configuration examples; for more details about the syntax of commands, see the *OmniSwitch CLI Reference Guide*.

Configuration procedures described in this chapter include:

- Defining DHCP rules on page 8-12.
- Defining binding rules to restrict access to specific network devices on page 8-14.
- Defining MAC address rules on page 8-17.
- Defining IP and IPX network address rules on page 8-18.
- Defining protocol rules on page 8-20.
- Defining user-defined (custom) rules on page 8-21.
- Defining forwarding-only port rules on page 8-21.
- Verifying the VLAN rule configuration on page 8-25.

For information about creating and managing VLANs, see Chapter 5, "Configuring VLANs."

For information about enabling port mobility and defining mobile port properties, see Chapter 7, "Assigning Ports to VLANs."

VLAN Rules Specifications

IEEE Standards Supported	802.1Q–*Virtual Bridged Local Area Networks* 802.1v–*VLAN Classification by Protocol and Port* 802.1D–*Media Access Control Bridges*
Maximum number of VLANs per switch	4094
Maximum number of rules per VLAN	Unlimited
Maximum number of rules per switch	8129 of each rule type, except for a DHCP generic rule because only one is allowed per switch.
Switch ports eligible for VLAN rule classification (dynamic VLAN assignment)	Mobile 10/100 Ethernet and gigabit ports.
Switch ports not eligible for VLAN rule classification	Non-mobile (fixed) ports. 802.1Q tagged fixed ports. Link aggregate ports.
CLI Command Prefix Recognition	All VLAN management commands support prefix recognition. See the "Using the CLI" chapter in the *OmniSwitch 7700/7800/8800 Switch Management Guide* for more information.

VLAN Rules Defaults

Parameter Description	**Command**	**Default**
IP network address rule subnet mask	**vlan ip**	The IP address class range; Class A, B, or C.
IPX network address rule encapsulation	**vlan ipx**	Ethernet-II

Sample VLAN Rule Configuration

The following steps provide a quick tutorial that will create an IP network address and DHCP MAC range rule for VLAN 255, an IPX protocol rule for VLAN 355, and a MAC-IP-port binding rule for VLAN 1500. The remaining sections of this chapter provide further explanation of all VLAN rules and how they are defined.

1 Create VLAN 255 with a description (e.g., Finance IP Network) using the following command:

```
-> vlan 255 name "Finance IP Network"
```

2 Define an IP network address rule for VLAN 255 that will capture mobile port traffic containing a network 21.0.0.0 IP source address. For example:

```
-> vlan 255 ip 21.0.0.0
```

3 Define a DHCP MAC range rule for VLAN 255 that will capture mobile port DHCP traffic that contains a source MAC address that falls within the range specified by the rule. For example:

```
-> vlan 255 dhcp mac 00:DA:95:00:59:10 00:DA:95:00:59:9F
```

4 Define an IPX protocol rule for VLAN 355 that will capture mobile port traffic containing an IPX protocol type value. For example:

```
-> vlan 355 protocol ipx-e2
```

5 Define a MAC-IP-port binding rule that restricts assignment to VLAN 1500 to traffic received on mobile port 3/10 containing a MAC address of 00:DA:95:00:CE:3F and an IP address of 21.0.0.43. For example:

```
-> vlan 1500 binding mac-ip-port 00:da:95:00:ce:3f 21.0.0.43 3/10
```

Note. *Optional*. To verify that the rules in this tutorial were defined for VLANs 255, 355, and 1500, enter **show vlan rules.** For example:

```
-> show vlan rules

Legend: type: * = binding rule

  type             vlan    rule
-----------------+------+-------------------------------------------------------
  ip-net            255    21.0.0.0, 255.0.0.0
  protocol          355    ipx-e2
  mac-ip-port*     1500    00:da:95:00:ce:3f, 21.0.0.43, 3/10
  dhcp-mac-range    255    00:da:95:00:59:10, 00:da:95:00:59:9f
```

VLAN Rules Overview

The mobile port feature available on the switch allows dynamic VLAN port assignment based on VLAN rules that are applied to mobile port traffic.When a port is defined as a mobile port, switch software compares traffic coming in on that port with configured VLAN rules. If any of the mobile port traffic matches any of the VLAN rules, the port and the matching traffic become a member of that VLAN.

VLANs do not have a mobile or non-mobile distinction and there is no overall switch setting to invoke the mobile port feature. Instead, mobility is enabled on individual switch ports and rules are defined for individual VLANs to capture mobile port traffic. Refer to Chapter 7, "Assigning Ports to VLANs," for more information about using mobile ports and dynamic VLAN port assignments.

VLAN Rule Types

There are several types of configurable VLAN rules available for classifying different types of network device traffic. There is no limit to the number of rules allowed per VLAN and up to 8,129 of each rule type is allowed per switch. See "Configuring VLAN Rule Definitions" on page 8-11 for instructions on how to create a VLAN rule.

The type of rule defined determines the type of traffic that will trigger a dynamic port assignment to the VLAN and the type of traffic the VLAN will forward within its domain. Refer to the following sections (listed in the order of rule precedence) for a description of each type of VLAN rule:

Rule	See
DHCP MAC Address DHCP MAC Range DHCP Port DHCP Generic	"DHCP Rules" on page 8-5
MAC-Port-IP Address Binding MAC-Port-Protocol Binding MAC-Port Binding MAC-IP Address Binding Port-IP Address Binding Port-Protocol Binding	"Binding Rules" on page 8-6
MAC Address MAC Address Range	"MAC Address Rules" on page 8-6
Network Address	"Network Address Rules" on page 8-6
Protocol	"Protocol Rules" on page 8-6
Custom	"Custom (User Defined) Rules" on page 8-7
Port	"Port Rules" on page 8-7

Use the **show vlan rules** command to display a list of rules already configured on the switch. For more information about this command, refer to the *OmniSwitch CLI Reference Guide*.

DHCP Rules

Dynamic Host Configuration Protocol (DHCP) frames are sent from client workstations to request an IP address from a DHCP server. The server responds with the same type of frames, which contain an IP address for the client. If clients are connected to mobile ports, DHCP rules are used to classify this type of traffic for the purposes of transmitting and receiving DHCP frames to and from the server.

When a mobile port receives a DHCP frame that matches a DHCP rule, the port is temporarily assigned to the VLAN long enough to forward the DHCP requests within the VLAN broadcast domain. The source MAC address of the DHCP frame, however, is not learned for that VLAN port association. As a result, the **show mac-address-table** command output will not contain an entry for the DHCP source MAC address. The **show vlan port** command output, however, will contain an entry for the temporary VLAN port association that occurs during this process.

Once a device connected to a mobile port receives an IP address from the DHCP server, the VLAN port assignment triggered by the device's DHCP frames matching a VLAN DHCP rule is dropped unless regular port traffic matches another rule on that same VLAN. If this match occurs, or the traffic matches a rule on another VLAN, then the source MAC address of the mobile port's frames is learned for that VLAN port association.

DHCP rules are most often used in combination with IP network address rules. A DHCP client has an IP address of all zeros (0.0.0.0) until it receives an IP address from a DHCP server, so initially it would not match any IP network address rules.

Binding rules, MAC address rules, and protocol rules also capture DHCP client traffic. The exception to this is binding rules that specify an IP address as part of the rule, similar to IP network address rule definitions.

The following DHCP rule types are available:

- DHCP MAC Address
- DHCP MAC Range
- DHCP Port
- DHCP Generic

Binding Rules

Binding rules restrict VLAN assignment to specific devices by requiring that device traffic match all criteria specified in the rule. As a result, a separate binding rule is required for each device. An unlimited number of such rules, however, is allowed per VLAN and up to 8,129 of each rule type is allowed per switch. Although DHCP traffic is examined and processed first by switch software, binding rules take precedence over all other rules.

The following binding rule types are available. The rule type name indicates the criteria the rule uses to determine if device traffic qualifies for VLAN assignment. For example, the MAC-Port-IP address binding rule requires a matching source MAC and IP address in frames received from a device connected to the port specified in the rule.

- MAC-port-IP Address
- MAC-port-protocol
- MAC-port
- MAC-IP Address
- port-IP address
- port-protocol

Note that MAC-port-IP, MAC-port-protocol, MAC-port, and port-IP binding rules are also supported on Authenticated VLANs (AVLANs). See "Configuring VLAN Rule Definitions" on page 8-11 and Chapter 21, "Configuring Authenticated VLANs," for more information.

MAC Address Rules

MAC address rules determine VLAN assignment based on a device's source MAC address. This is the simplest type of rule and provides the maximum degree of control and security. Members of the VLAN will consist of devices with specific MAC addresses. In addition, once a device joins a MAC address rule VLAN, it is not eligible to join multiple VLANs even if device traffic matches other VLAN rules.

MAC address rules also capture DHCP traffic, if no other DHCP rule exists that would classify the DHCP traffic into another VLAN. Therefore, it is not necessary to combine DHCP rules with MAC address rules for the same VLAN.

Network Address Rules

There are two types of network address rules: IP and IPX. An IP network address rule determines VLAN mobile port assignment based on a device's source IP address. An IPX network address rule determines VLAN mobile port assignment based on a device's IPX network and encapsulation.

Protocol Rules

Protocol rules determine VLAN assignment based on the protocol a device uses to communicate. When defining this type of rule, there are several generic protocol values to select from: IP, IPX, AppleTalk, or DECNet. If none of these are sufficient, it is possible to specify an Ethernet type, Destination and Source Service Access Protocol (DSAP/SSAP) header values, or a Sub-network Access Protocol (SNAP) type.

Note that specifying a SNAP protocol type restricts classification of mobile port traffic to the ethertype value found in the IEEE 802.2 SNAP LLC frame header.

IP protocol rules also capture DHCP traffic, if no other DHCP rule exists that would classify the DHCP traffic into another VLAN. Therefore, it is not necessary to combine DHCP rules with IP protocol rules for the same VLAN.

Custom (User Defined) Rules

Custom rules determine VLAN assignment based on criteria defined by the user. The criteria consists of a specified pattern of data and a location where that data must exist within the frame. Devices originating frames that contain this same data in the required frame location are dynamically assigned to the VLAN.

Defining a custom rule is recommended only if none of the other available rules provide the necessary criteria for capturing the desired type of mobile port traffic.

Port Rules

Port rules are fundamentally different from all other supported rule types, in that traffic is not required to trigger dynamic assignment of the mobile port to a VLAN. As soon as this type of rule is created, the specified port is assigned to the VLAN only for the purpose of forwarding broadcast types of VLAN traffic to a device connected to that same port.

Port rules are mostly used for silent devices, such as printers, that require VLAN membership to receive traffic forwarded from the VLAN. These devices usually don't send traffic, so they do not trigger dynamic assignment of their mobile ports to a VLAN.

It is also possible to specify the same port in more than one port rule defined for different VLANs. The advantage to this is that traffic from multiple VLANs is forwarded out the one mobile port to the silent device. For example, if port 3 on slot 2 is specified in a port rule defined for VLANs 255, 355, and 755, then outgoing traffic from all three of these VLANs is forwarded on port 2/3.

Port rules only apply to outgoing mobile port traffic and do not classify incoming traffic. If a mobile port is specified in a port rule, its incoming traffic is still classified for VLAN assignment in the same manner as all other mobile port traffic.

VLAN assignments that are defined using port rules are exempt from the port's default VLAN restore status. See Chapter 7, "Assigning Ports to VLANs," for more information regarding a port's default VLAN restore status and other mobile port properties.

Understanding VLAN Rule Precedence

In addition to configurable VLAN rule types, there are two internal rule types for processing mobile port frames. One is referred to as *frame type* and is used to identify Dynamic Host Configuration Protocol (DHCP) frames. The second internal rule is referred to as *default* and identifies frames that do not match any VLAN rules.

The VLAN rule precedence table on page 8-8 provides a list of all VLAN rules, including the two internal rules mentioned above, in the order of precedence switch software applies to classify mobile port frames. The first column lists the rule type names, the second and third columns describe how the switch handles frames that match or don't match rule criteria. The higher the rule is in the list, the higher its level of precedence.

When a frame is received on a mobile port, switch software starts with rule one in the rule precedence table and progresses down the list until there is a successful match between rule criteria and frame contents. The exception to this is if there is a binding rule violation. In this case, the frame is blocked and its source port is not assigned to the rule's VLAN.

Each binding rule type contains criteria that is used to determine if a mobile port frame qualifies for assignment to the binding rule VLAN, violates binding rule criteria, or is simply allowed on the port but not assigned to the rule's VLAN. For example, as indicated in the rule precedence table, a mobile port frame is compared to binding MAC-port rule criteria and processed as follows:

- If the frame's source MAC address matches the rule's MAC address, then the frame's port must also match the rule's port to qualify for assignment to the rule's VLAN.
- If the frame's source MAC matches but the frame's port does *not* match, then a violation occurs and the frame is blocked and the port is not assigned to the rule's VLAN. There is no further attempt to match this frame to rules of lower precedence.
- If the frame's source MAC does not match but the frame's port does match, the frame is allowed but the port is not assigned to the rule's VLAN. The frame is then compared to other rules of lower precendence in the table or carried on the mobile port's default VLAN if the frame does not match any other VLAN rules and the mobile port's default VLAN is enabled.

Precedence Step/Rule Type	**Condition**	**Result**
1. Frame Type	Frame is a DHCP frame.	Go to Step 2.
	Frame is not a DHCP frame.	Skip Steps 2, 3, 4, and 5.
2. DHCP MAC	DHCP frame contains a matching source MAC address.	Frame source is assigned to the rule's VLAN, but not learned.
3. DHCP MAC Range	DHCP frame contains a source MAC address that falls within a specified range of MAC addresses.	Frame source is assigned to the rule's VLAN, but not learned.
4. DHCP Port	DHCP frame matches the port specified in the rule.	Frame source is assigned to the rule's VLAN, but not learned.
5. DHCP Generic	DHCP frame.	Frame source is assigned to the rule's VLAN, but not learned.
6. MAC-Port-IP Address Binding	Frame contains a matching source MAC address, source port, and source IP subnet address.	Frame source is assigned to the rule's VLAN.
	Frame only contains a matching source MAC address; port and IP address do not match.	Frame is blocked; its source is not assigned to the rule's VLAN.
	Frame only contains a matching IP address; source MAC and port do not match.	Frame is blocked; its source is not assigned to the rule's VLAN.
	Frame only contains a matching port; source MAC and IP address do not match.	Frame is allowed; its source is not assigned to the rule's VLAN.

Precedence Step/Rule Type	Condition	Result
7. MAC-Port-Protocol Binding	Frame contains a matching source MAC address, source port, and protocol.	Frame source is assigned to the rule's VLAN.
	Frame only contains a matching source MAC address; port and protocol do not match.	Frame is blocked; its source is not assigned to the rule's VLAN.
	Frame only contains a matching port and/or protocol; source MAC address does not match.	Frame is allowed; its source is not assigned to the rule's VLAN.
8. MAC-Port Binding	Frame contains a matching source MAC address and source port.	Frame source is assigned to the rule's VLAN.
	Frame only contains a matching source MAC address; port does not match.	Frame is blocked; its source is not assigned to the rule's VLAN.
	Frame only contains a matching port; source MAC address does not match.	Frame is allowed; its source is not assigned to the rule's VLAN.
9. MAC-IP Address Binding	Frame contains a matching source MAC address and source IP subnet address.	Frame source is assigned to the rule's VLAN.
	Frame only contains a matching source MAC address; IP address does not match.	Frame is blocked; its source is not assigned to the rule's VLAN.
	Frame only contains a matching IP address; source MAC does not match.	Frame is blocked; its source is not assigned to the rule's VLAN.
10. Port-IP Address Binding	Frame contains a matching source port and source IP subnet address.	Frame source is assigned to the rule's VLAN.
	Frame only contains a matching source IP address; port does not match.	Frame is blocked; its source is not assigned to the rule's VLAN.
	Frame only contains a matching port; source IP address does not match.	Frame is allowed; its source is not assigned to the rule's VLAN.

Precedence Step/Rule Type	Condition	Result
11. Port-Protocol Binding	Frame contains a matching source port and protocol. Frame only contains a matching source port; protocol does not match. Frame only contains a matching protocol; port does not match.	Frame source is assigned to the rule’s VLAN. Frame is blocked; its source is not assigned to the rule’s VLAN. Frame is allowed; its source is not assigned to the rule’s VLAN.
12. MAC Address	Frames contain a matching source MAC address.	Frame source is assigned to the rule’s VLAN.
13. MAC Range	Frame contains a source MAC address that falls within a specified range of MAC addresses.	Frame source is assigned to the rule’s VLAN.
14. Network Address	Frame contains a matching IP subnet address, or Frame contains a matching IPX network address.	Frame source is assigned to the rule’s VLAN. Frame source is assigned to the rule’s VLAN.
15. Protocol	Frame contains a matching protocol type.	Frame source is assigned to the rule’s VLAN.
16. Custom (User Defined)	Frames contain data that matches customized rule criteria.	Frame source is assigned to the rule’s VLAN.
17. Default	Frame does not match any rules.	Frame source is assigned to mobile port’s default VLAN.

Configuring VLAN Rule Definitions

Consider the following when configuring rules for a VLAN:

- The VLAN must already exist. Use the **vlan** command to create a new VLAN or the **show vlan** command to verify a VLAN is already configured. Refer to Chapter 5, "Configuring VLANs," for more information.
- Which type of rule to define; DHCP, binding, MAC address, protocol, network address, custom, or port. Refer to "VLAN Rule Types" on page 8-4 for a summary of rule type definitions.
- What is the rule's precedence compared to other rules defined for other VLANs. If mobile port traffic matches rules defined for more than one VLAN, the mobile port is dynamically assigned to the VLAN with the higher precedence rule. Refer to "Understanding VLAN Rule Precedence" on page 8-7 for more information.
- It is possible to define multiple rules for the same VLAN, as long as each rule is different. If mobile port traffic matches only one of the rules, the port and traffic are dynamically assigned to that VLAN.
- There is no limit to the number of rules defined for a single VLAN and up to 8,129 of each rule type is allowed per switch.
- It is possible to create a custom rule or protocol rules based on Ether type, SNAP type, or DSAP/SSAP values. It is recommended, however, to use predefined rules (such as MAC address, network address, and generic protocol rules) whenever possible to ensure accurate results when capturing mobile port traffic.
- When a VLAN is administratively disabled, static port and dynamic mobile port assignments are retained but traffic on these ports is not forwarded. However, VLAN rules remain active and continue to classify mobile port traffic for VLAN membership.
- When a VLAN is deleted from the switch configuration, all rules defined for that VLAN are automatically removed and any static or dynamic port assignments are dropped.
- It is possible to define MAC-port-IP, MAC-port-protocol, MAC-port, and port-IP binding rules for Authenticated VLANs (AVLANs). However, these rules are not active until the **avlan port-bound** command is issued for the AVLAN. Note that these rules only apply to traffic received on authenticated ports. See Chapter 21, "Configuring Authenticated VLANs," for more information.

Refer to the following sections (listed in the order of rule precedence) for instructions on how to define each type of VLAN rule:

Rule	See
DHCP MAC Address	“Defining DHCP MAC Address Rules” on page 8-12
DHCP MAC Range	“Defining DHCP MAC Range Rules” on page 8-13
DHCP Port	“Defining DHCP Port Rules” on page 8-13
DHCP Generic	“Defining DHCP Generic Rules” on page 8-14
MAC-Port-IP Address Binding MAC-Port-Protocol Binding MAC-Port Binding MAC-IP Address Binding Port-IP Address Binding Port-Protocol Binding	“Defining Binding Rules” on page 8-14
MAC Address	“Defining MAC Address Rules” on page 8-17
MAC Address Range	“Defining MAC Range Rules” on page 8-18
Network Address	“Defining IP Network Address Rules” on page 8-18 and “Defining IPX Network Address Rules” on page 8-19
Protocol	“Defining Protocol Rules” on page 8-20
Custom	“Defining Custom (User) Rules” on page 8-21
Port	“Defining Port Rules” on page 8-21

To display a list of VLAN rules already configured on the switch, use the **show vlan rules** command. For more information about this command, refer to the *OmniSwitch CLI Reference Guide*.

Defining DHCP MAC Address Rules

DHCP MAC address rules capture DHCP frames that contain a source MAC address that matches the MAC address specified in the rule. See “Application Example: DHCP Rules” on page 8-22 for an example of how DHCP port rules are used in a typical network configuration.

To define a DHCP MAC address rule, enter **vlan** followed by an existing VLAN ID then **dhcp mac** followed by a valid MAC address. For example, the following command defines a DHCP MAC address rule for VLAN 255:

```
-> vlan 255 dhcp mac 00:00:da:59:0c:11
```

Only one MAC address is specified when using the **vlan dhcp mac** command to create a DHCP MAC rule. Therefore, to specify multiple MAC addresses for the same VLAN, create a DHCP MAC rule for each address. If dealing with a large number of MAC addresses in sequential order, consider using a DHCP MAC range rule described in the next section.

Use the **no** form of the **vlan dhcp mac** command to remove a DHCP MAC address rule.

```
-> vlan 255 no dhcp mac 00:00:da:59:0c:11
```

Defining DHCP MAC Range Rules

A DHCP MAC range rule is similar to a DHCP MAC address rule, but allows the user to specify a range of MAC addresses. This is useful when it is necessary to define rules for a large number of sequential MAC addresses. One DHCP MAC range rule could serve the same purpose as 10 or 20 DHCP MAC address rules, requiring less work to configure.

DHCP frames that contain a source MAC address that matches the low or high end MAC or that falls within the range specified by the low and high end MAC trigger dynamic port assignment to the rule's VLAN. To define a DHCP MAC range rule, enter **vlan** followed by an existing VLAN ID then **dhcp mac range** followed by valid low and high end MAC addresses. For example, the following command creates a DHCP MAC range rule for VLAN 1100:

```
-> vlan 1100 dhcp mac range 00:00:da:00:00:01 00:00:da:00:00:09
```

Only valid source MAC addresses are allowed for the low and high end boundary MACs. For example, multicast addresses (e.g., 01:00:00:c5:09:1a) are ignored even if they fall within a specified MAC range and are not allowed as the low or high end boundary MAC. If an attempt is made to use a multicast address for one of the boundary MACs, an error message is displayed and the rule is not created.

Use the **no** form of the **vlan dhcp mac range** command to remove a DHCP MAC range rule. Note that it is only necessary to enter the low end MAC address to identify which rule to remove.

```
-> vlan 1000 no dhcp mac range 00:00:da:00:00:01
```

Defining DHCP Port Rules

DHCP port rules capture DHCP frames that are received on a mobile port that matches the port specified in the rule. See "Application Example: DHCP Rules" on page 8-22 for an example of how DHCP port rules are used in a typical network configuration.

To define a DHCP port rule, enter **vlan** followed by an existing VLAN ID then **dhcp port** followed by a *slot/port* designation. For example, the following command defines a DHCP port rule for VLAN 255:

```
-> vlan 255 dhcp port 2/3
```

To specify multiple ports and/or slots, use a hyphen to specify a range of ports and a space to specify multiple slots. For example,

```
-> vlan 255 dhcp port 4/1-5 5/12-20 6/10-15
```

Use the **no** form of the **vlan dhcp port** command to remove a DHCP port rule.

```
-> vlan 255 no dhcp port 2/10-12 3/1-5 6/1-9
```

Defining DHCP Generic Rules

DHCP generic rules capture all DHCP traffic that does not match an existing DHCP MAC or DHCP port rule. If none of these other rules exist, then all DHCP frames are captured regardless of the port they came in on or the frame's source MAC address. Only one rule of this type is allowed per switch.

To define a DHCP generic rule, enter **vlan** followed by an existing VLAN ID then **dhcp generic.** For example,

```
-> vlan 255 dhcp generic
```

Use the **no** form of the **vlan dhcp generic** command to remove a DHCP generic rule.

```
-> vlan 255 no dhcp generic
```

Defining Binding Rules

If a binding rule is defined for a VLAN, then mobile port traffic must match all criteria of the binding rule to qualify for dynamic assignment to that VLAN. Binding rule criteria consists of one of six combinations, each of which is a specific binding rule type:

1 The device must attach to a specific switch port *and* use a specific MAC address *and* use a specific IP network address (MAC-port-IP address binding rule).

2 The device must attach to a specific switch port *and* use a specific source MAC address *and* use a specific protocol (MAC-port-Protocol binding rule).

3 The device must use a specific port *and* a specific source MAC address (MAC-port binding rule).

4 The device must use a specific IP address *and* use a specific MAC address (MAC-IP address binding rule).

5 The device must use a specific port *and* a specific IP address (port-IP address binding rule).

6 The device must attach to a specific switch port *and* use a specific protocol (port-protocol binding rule).

If frames do not contain matching criteria, they are compared against other existing VLAN rules of lower precedence. However, if a frame violates criteria of any one binding rule, it is discarded. Refer to "Understanding VLAN Rule Precedence" on page 8-7 for more information.

Note that MAC-port-IP, MAC-port-Protocol, MAC-port, and port-IP binding rules are also supported on Authenticated VLANs (AVLANs). See Chapter 21, "Configuring Authenticated VLANs," for more information.

The following subsections provide information about how to define each of the binding rule types.

How to Define a MAC-Port-IP Address Binding Rule

To define a MAC-port-IP address binding rule, enter **vlan** followed by an existing VLAN ID then **binding mac-ip-port** followed by a valid MAC address, IP address, and a *slot/port* designation. For example, the following command defines a MAC-port-IP binding rule for VLAN 255:

```
-> vlan 255 binding mac-ip-port 00:00:da:59:0c:12 21.0.0.10 2/3
```

In this example, frames received on mobile port 2/3 must contain a source MAC address of 00:00:da:59:0c:12 and a source IP address of 21.0.0.10 to qualify for dynamic assignment to VLAN 255.

Use the **no** form of the **vlan binding mac-ip-port** command to remove a MAC-port-IP binding rule. Note that it is only necessary to enter the rule's MAC address parameter value to identify which rule to remove.

```
-> vlan 255 no binding mac-ip-port 00:00:da:59:0c:12
```

Note that this binding rule type is also supported on AVLANs. See Chapter 21, "Configuring Authenticated VLANs," for more information.

How to Define a MAC-Port-Protocol Binding Rule

To define a MAC-port-protocol binding rule, enter **vlan** followed by an existing VLAN ID then **binding mac-port-protocol** followed by a valid MAC address, a *slot/port* designation and a protocol type. For example, the following commands define a MAC-port-protocol binding rule for VLAN 355 and VLAN 455:

```
-> vlan 355 binding mac-port-protocol 00:00:da:59:0c:12 3/1 ip-e2
-> vlan 455 binding mac-port-protocol 00:00:20:11:4a:29 4/1 dsapssap 04/04
```

The first example command specifies that frames received on mobile port 3/1 must contain a source MAC address of 00:00:da:59:0c:12 and an IP protocol type to qualify for dynamic assignment to VLAN 355. The second command specifies that frames received on mobile port 4/1 must contain a source MAC address of 00:00:20:11:4a:29 and a DSAP/SSAP protocol value of 04/04 to qualify for dynamic assignment to VLAN 455.

The following table lists command keywords for specifying a protocol type:

protocol type keywords	
ip-e2	**decnet**
ip-snap	**appletalk**
ipx-e2	**ethertype**
ipx-novell	**dsapssap**
ipx-llc	**snap**
ipx-snap	

Note that specifying a SNAP protocol type restricts classification of mobile port traffic to the ethertype value found in the IEEE 802.2 SNAP LLC frame header.

Use the **no** form of the **vlan binding mac-port-protocol** command to remove a MAC-port-protocol binding rule. Note that it is only necessary to enter the rule's MAC address and protocol parameter values to identify which rule to remove.

```
-> vlan 455 no binding mac-port-protocol 00:00:20:11:4a:29 dsapssap 04/04
```

Note that this binding rule type is also supported on AVLANs. See Chapter 21, "Configuring Authenticated VLANs," for more information.

How to Define a MAC-Port Binding Rule

To define a MAC-port binding rule, enter **vlan** followed by an existing VLAN ID then **binding mac-port** followed by a valid MAC address and a *slot/port* designation. For example, the following command defines a MAC-port binding rule for VLAN 1500:

```
-> vlan 1500 binding mac-port 00:02:9a:3e:f1:06 6/10
```

In this example, frames received on mobile port 6/10 must contain a source MAC address of 00:02:9a:3e:f1:06 to qualify for dynamic assignment to VLAN 1500.

Use the **no** form of the **vlan binding mac-port** command to remove a MAC-port binding rule. Note that it is only necessary to enter the rule's MAC address parameter value to identify which rule to remove.

```
-> vlan 1500 no binding mac-port 00:02:9a:3e:f1:06
```

Note that this binding rule type is also supported on AVLANs. See Chapter 21, "Configuring Authenticated VLANs," for more information.

How to Define a MAC-IP Address Binding Rule

To define a MAC-IP address binding rule, enter **vlan** followed by an existing VLAN ID then **binding mac-ip** followed by a valid IP subnet address. For example, the following command defines a MAC-IP binding rule for VLAN 1501:

```
-> vlan 1501 binding mac-ip 00:02:9a:3e:f1:07 172.16.6.3
```

In this example, frames received on any mobile port must contain a source MAC address of 00:02:9a:3e:f1:07 and a source IP subnet address of 172.16.6.3 to qualify for dynamic assignment to VLAN 1501.

Use the **no** form of the **vlan binding mac-ip** command to remove a MAC-IP binding rule. Note that it is only necessary to enter the rule's MAC address parameter value to identify which rule to remove.

```
-> vlan 1500 no binding mac-port 00:02:9a:3e:f1:07
```

How to Define an IP-Port Binding Rule

To define a IP-port binding rule, enter **vlan** followed by an existing VLAN ID then **binding ip-port** followed by a valid IP subnet address and a *slot/port* designation. For example, the following command defines an IP-port binding rule for VLAN 1502:

```
-> vlan 1502 binding ip-port 172.16.6.4 5/12
```

In this example, frames received on mobile port 5/12 must contain a source IP subnet address of 172.16.6.4 to qualify for dynamic assignment to VLAN 1502.

Use the **no** form of the **vlan binding ip-port** command to remove an IP-port binding rule. Note that it is only necessary to enter the rule's IP subnet address parameter value to identify which rule to remove.

```
-> vlan 1502 no binding ip-port 172.16.6.4
```

Note that this binding rule type is also supported on AVLANs. See Chapter 21, "Configuring Authenticated VLANs," for more information.

How to Define a Port-Protocol Binding Rule

To define a port-protocol binding rule, enter **vlan** followed by an existing VLAN ID then **binding port-protocol** followed by a valid MAC address, a *slot/port* designation and a protocol type. For example, the following commands define a port-protocol binding rule for VLAN 1503 and VLAN 1504:

```
-> vlan 1503 binding port-protocol 3/1 ip-snap
-> vlan 1504 binding port-protocol 4/1 dsapssap F0/F0
```

The first example command specifies that frames received on mobile port 3/1 must contain an IP SNAP protocol type to qualify for dynamic assignment to VLAN 1503. The second command specifies that frames received on mobile port 4/1 must contain a DSAP/SSAP protocol value of F0/F0 to qualify for dynamic assignment to VLAN 1504.

The following table lists command keywords for specifying a protocol type:

protocol type keywords	
ip-e2	**decnet**
ip-snap	**appletalk**
ipx-e2	**ethertype**
ipx-novell	**dsapssap**
ipx-llc	**snap**
ipx-snap	

Note that specifying a SNAP protocol type restricts classification of mobile port traffic to the ethertype value found in the IEEE 802.2 SNAP LLC frame header.

Use the **no** form of the **vlan binding port-protocol** command to remove a port-protocol binding rule.

```
-> vlan 255 no binding port-protocol 8/12 ethertype 0600
```

Defining MAC Address Rules

MAC address rules capture frames that contain a source MAC address that matches the MAC address specified in the rule. The mobile port that receives the matching traffic is dynamically assigned to the rule's VLAN. Using MAC address rules, however, limits dynamic port assignment to a single VLAN. A mobile port can only belong to one MAC address rule VLAN, even if it sends traffic that matches rules defined for other VLANs.

For example, if VLAN 10 has a MAC address rule defined for 00:00:2a:59:0c:f1 and VLAN 20 has an IP protocol rule defined, mobile port 4/2 sending IP traffic with a source MAC address of 00:00:2a:59:0c:f1 is only assigned to VLAN 10. All mobile port 4/2 traffic is forwarded on VLAN 10, even though its traffic also matches the VLAN 20 IP protocol rule.

To define a MAC address rule, enter **vlan** followed by an existing VLAN ID then **mac** followed by a valid MAC address. For example, the following command defines a MAC address rule for VLAN 255:

```
-> vlan 255 mac 00:00:da:59:0c:11
```

Only one MAC address is specified when using the **vlan mac** command to create a MAC address rule. Therefore, to specify multiple MAC addresses for the same VLAN, create a separate rule for each address. If dealing with a large number of MAC addresses, consider using MAC address range rules described in the next section.

Use the **no** form of the **vlan mac** command to remove a MAC address rule.

```
-> vlan 255 no mac 00:00:da:59:0c:11
```

Defining MAC Range Rules

A MAC range rule is similar to a MAC address rule, but allows the user to specify a range of MAC addresses. This is useful when it is necessary to define rules for a large number of sequential MAC addresses. One MAC range rule could serve the same purpose as 10 or 20 MAC address rules, requiring less work to configure.

Frames that contain a source MAC address that matches the low or high end MAC or that falls within the range specified by the low and high end MAC trigger dynamic port assignment to the rule's VLAN. As is the case with MAC address rules, dynamic port assignment is limited to a single VLAN. A mobile port can only belong to one MAC range rule VLAN, even if it sends traffic that matches rules defined for other VLANs.

To define a MAC range rule, enter **vlan** followed by an existing VLAN ID then **mac range** followed by valid low and high end MAC addresses. For example, the following command creates a MAC range rule for VLAN 1000:

```
-> vlan 1000 mac range 00:00:da:00:00:01 00:00:da:00:00:09
```

Only valid source MAC addresses are allowed for the low and high end boundary MACs. For example, multicast addresses (e.g., 01:00:00:c5:09:1a) are ignored even if they fall within a specified MAC range and are not allowed as the low or high end boundary MAC. If an attempt is made to use a multicast address for one of the boundary MACs, an error message is displayed and the rule is not created.

Use the **no** form of the **vlan mac range** command to remove a MAC range rule. Note that it is only necessary to enter the low end MAC address to identify which rule to remove.

```
-> vlan 1000 no mac range 00:00:da:00:00:01
```

Defining IP Network Address Rules

IP network address rules capture frames that contain a source IP subnet address that matches the IP subnet address specified in the rule. If DHCP is used to provide client workstations with an IP address, consider using one of the DHCP rules in combination with an IP network address rule. See "Application Example: DHCP Rules" on page 8-22 for an example of how IP network address and DHCP rules are used in a typical network configuration.

To define an IP network address rule, enter **vlan** followed by an existing VLAN ID then **ip** followed by a valid IP network address and an optional subnet mask. For example, the following command creates an IP network address rule for VLAN 1200:

```
-> vlan 1200 ip 31.0.0.0 255.0.0.0
```

In this example, frames received on any mobile port must contain a network 31.0.0.0 source IP address (e.g., 31.0.0.10, 31.0.0.4) to qualify for dynamic assignment to VLAN 1200.

If a subnet mask is not specified, the default class for the IP address is used (Class A, B, or C). For example, either one of the following commands will create an IP network address rule for network 134.10.0.0:

```
-> vlan 1200 ip 134.10.0.0 255.255.0.0
-> vlan 1200 ip 134.10.0.0
```

The pool of available internet IP addresses is divided up into three classes, as shown in the following table. Each class includes a range of IP addresses. The range an IP network address belongs to determines the default class for the IP network when a subnet mask is not specified.

Network Range	Class
1.0.0.0 - 126.0.0.0	A
128.1.0.0 - 191.254.0.0	B
192.0.1.0 - 223.255.254.0	C

Use the **no** form of the **vlan ip** command to remove an IP network address rule.

```
-> vlan 1200 no ip 134.10.0.0
```

Defining IPX Network Address Rules

IPX network address rules capture frames that contain an IPX network address and encapsulation that matches the IPX network and encapsulation specified in the rule. This rule only applies to devices that already have an IPX network address assigned.

To define an IPX network address rule, enter **vlan** followed by an existing VLAN ID then **ipx** followed by a valid IPX network number and an optional encapsulation parameter value. For example, the following command creates an IPX network address rule for VLAN 1200:

```
-> vlan 1200 ipx a010590c novell
```

In this example, frames received on any mobile port must contain an IPX network a010590c address with a Novell Raw (802.3) encapsulation to qualify for dynamic assignment to VLAN 1200.

IPX network addresses consist of eight hex digits. If an address less than eight digits is entered, the entry is prefixed with zeros to equal eight characters. For example, the following command results in an IPX network address rule for network 0000250b:

```
-> vlan 1210 ipx 250b snap
```

If an encapsulation parameter value is not specified, this value defaults to Ethernet-II encapsulation. For example, either one of the following commands creates the same IPX network address rule:

```
-> vlan 1220 ipx 250c e2
-> vlan 1220 ipx 250c
```

If the IPX network address rule VLAN is going to route IPX traffic, it is important to specify a rule encapsulation that matches the IPX router port encapsulation. If there is a mismatch, connectivity with other IPX devices may not occur. See Chapter 5, “Configuring VLANs,” for information about defining VLAN IPX router ports.

The following table lists keywords for specifying an encapsulation value:

IPX encapsulation keywords	
e2 or ethernet2 **llc**	**snap** **novell**

Use the **no** form of the **vlan ipx** command to remove an IPX network address rule. Note that it is only necessary to specify the IPX network address to identify which rule to remove:

```
-> vlan 1220 no ipx 250c
```

Defining Protocol Rules

Protocol rules capture frames that contain a protocol type that matches the protocol value specified in the rule. There are several generic protocol parameter values to select from; IP Ethernet-II, IP SNAP, IPX Ethernet II, IPX Novell (802.3), IPX LLC (802.2), IPX SNAP, DECNet, and Appletalk. If none of these are sufficient to capture the desired type of traffic, use the Ethertype, DSAP/SSAP, or SNAP parameters to define a more specific protocol type value.

To define a protocol rule, enter **vlan** followed by an existing VLAN ID then **protocol** followed by a valid protocol parameter value. For example, the following commands define a protocol rule for VLAN 1503 and VLAN 1504:

```
-> vlan 1503 protocol ip-snap
-> vlan 1504 protocol dsapssap f0/f0
```

The first example command specifies that frames received on any mobile port must contain an IP SNAP protocol type to qualify for dynamic assignment to VLAN 1503. The second command specifies that frames received on any mobile port must contain a DSAP/SSAP protocol value of f0/f0 to qualify for dynamic assignment to VLAN 1504.

If an attempt is made to define an Ethertype rule with a protocol type value that is equal to the value already captured by one of the generic IP or IPX protocol rules, a message displays recommending the use of the IP or IPX generic rule. The following example shows what happens when an attempt is made to create a protocol rule with an Ethertype value of 0800 (IP Ethertype):

```
-> vlan 200 protocol ethertype 0800
ERROR: Part of ip ethernet protocol class - use <vlan # protocol ip-e2> instead
```

The following table lists keywords for specifying a protocol type:

protocol type keywords	
ip-e2	**decnet**
ip-snap	**appletalk**
ipx-e2	**ethertype**
ipx-novell	**dsapssap**
ipx-llc	**snap**
ipx-snap	

Note that specifying a SNAP protocol type restricts classification of mobile port traffic to the ethertype value found in the IEEE 802.2 SNAP LLC frame header.

Use the **no** form of the **vlan protocol** command to remove a protocol rule.

```
-> vlan 1504 no protocol dsapssap f0/f0
```

Defining Custom (User) Rules

A custom rule captures mobile port frames that contain a specified pattern of data at a specified location. Custom rules require the user to specify the following parameter values:

Parameter	Definition
offset	A number between 0 and 72. Specifies the number of bytes into the frame where the pattern (value) is found.
value	A four byte hex value that specifies a pattern of data (e.g., 60020000).
mask	A four byte hex value that identifies the bytes in the pattern to compare to the frame contents at the offset location. Use any hex character in the *mask* to mark bytes in the pattern to match and '0' to mark bytes in the pattern to ignore (e.g., eeff0000 is the *mask* for the 60020000 *value* pattern).

To define a custom rule, enter **vlan** followed by an existing VLAN ID then **user** followed by offset, data pattern, and mask values. For example, the following command creates a custom rule for VLAN 310:

```
-> vlan 310 user 14 e0000000 ab000000
```

In this example, frames received on a mobile port that contain **E0** in the specified data pattern located at the 14th byte of the frame qualify for dynamic assignment to VLAN 310.

Use the **no** form of the **vlan user** command to remove a custom rule. Note that it is only necessary to enter the offset and data pattern values to identify which rule to remove.

```
-> vlan 310 no user 14 e0000000
```

Defining Port Rules

Port rules do not require mobile port traffic to trigger dynamic assignment. When this type of rule is defined, the specified mobile port is immediately assigned to the specified VLAN. As a result, port rules are often used for silent network devices, which do not trigger dynamic assignment because they do not send traffic.

Port rules only apply to outgoing mobile port broadcast types of traffic and do not classify incoming traffic. In addition, multiple VLANs can have the same port rule defined. The advantage to this is that broadcast traffic from multiple VLANs is forwarded out one physical mobile port. When a mobile port is specified in a port rule, however, its incoming traffic is still classified for VLAN assignment in the same manner as all other mobile port traffic.

To define a port rule, enter **vlan** followed by an existing VLAN ID then **port** followed by the **slot/port** designation for a mobile port. For example, the following command creates a port rule for VLAN 755:

```
-> vlan 755 port 2/3
```

In this example, all traffic on VLAN 755 is flooded out mobile port 2 on slot 3.

Note that it is possible to define a port rule for a non-mobile (fixed, untagged) port, however, the rule is not active until mobility is enabled on the port.

Use the no form of the **vlan port** command to remove a port rule.

```
-> vlan 755 no port 2/3
```

Application Example: DHCP Rules

This application example shows how Dynamic Host Configuration Protocol (DHCP) port and MAC address rules are used in a DHCP-based network. DHCP is built on a client-server model in which a designated DHCP server allocates network addresses and delivers configuration parameters to dynamically configured clients.

Since DHCP clients initially have no IP address, assignment of these clients to a VLAN presents a problem. The switch determines VLAN membership by looking at traffic from source devices. Since the first traffic transmitted from a source DHCP client does not contain the actual address for the client (because the server has not allocated the address yet), the client may not have the same VLAN assignment as its server.

Before the introduction of DHCP port and MAC address rules, various strategies were deployed to use DHCP with VLANs. Typically these strategies involved IP protocol and network address rules along with DHCP Relay functionality. These solutions required the grouping of all DHCP clients in a particular VLAN through a common IP policy.

DHCP port and MAC address rules simplify the configuration of DHCP networks. Instead of relying on IP-based rules to group all DHCP clients in the same network as a DHCP server, you can manually place each individual DHCP client in the VLAN or mobile group of your choice.

Note. DHCP rules only provide temporary VLAN assignment to a client for the purposes of obtaining an IP address. Once the client receives an IP address from the server, traffic from that client is compared to other rules to determine the appropriate VLAN assignment for the client. If the traffic does not match any rules, it is forwarded on the default VLAN for the port.

The VLANs

This application example contains three (3) VLANs. These VLANs are called Test, Production, and Branch. The Test VLAN connects to the main network, the Production VLAN, through an external router. The configuration of this VLAN is self-contained, making it easy to duplicate for testing purposes. The Test VLAN contains its own DHCP server and DHCP clients. The clients gain membership to the VLAN through DHCP port rules.

The Production VLAN carries most of the traffic in this network. It does not contain a DHCP server, but does contain DHCP clients that gain membership through DHCP port rules. Two external routers connect this VLAN to the Test VLAN and a Branch VLAN. One of the external routers—the one connected to the Branch VLAN—has DHCP Relay functionality enabled. It is through this router that the DHCP clients in the Production VLAN access the DHCP server in the Branch VLAN.

The Branch VLAN contains a number of DHCP client stations and its own DHCP server. The DHCP clients gain membership to the VLAN through both DHCP port and MAC address rules. The DHCP server allocates IP addresses to all Branch and Production VLAN clients.

DHCP Servers and Clients

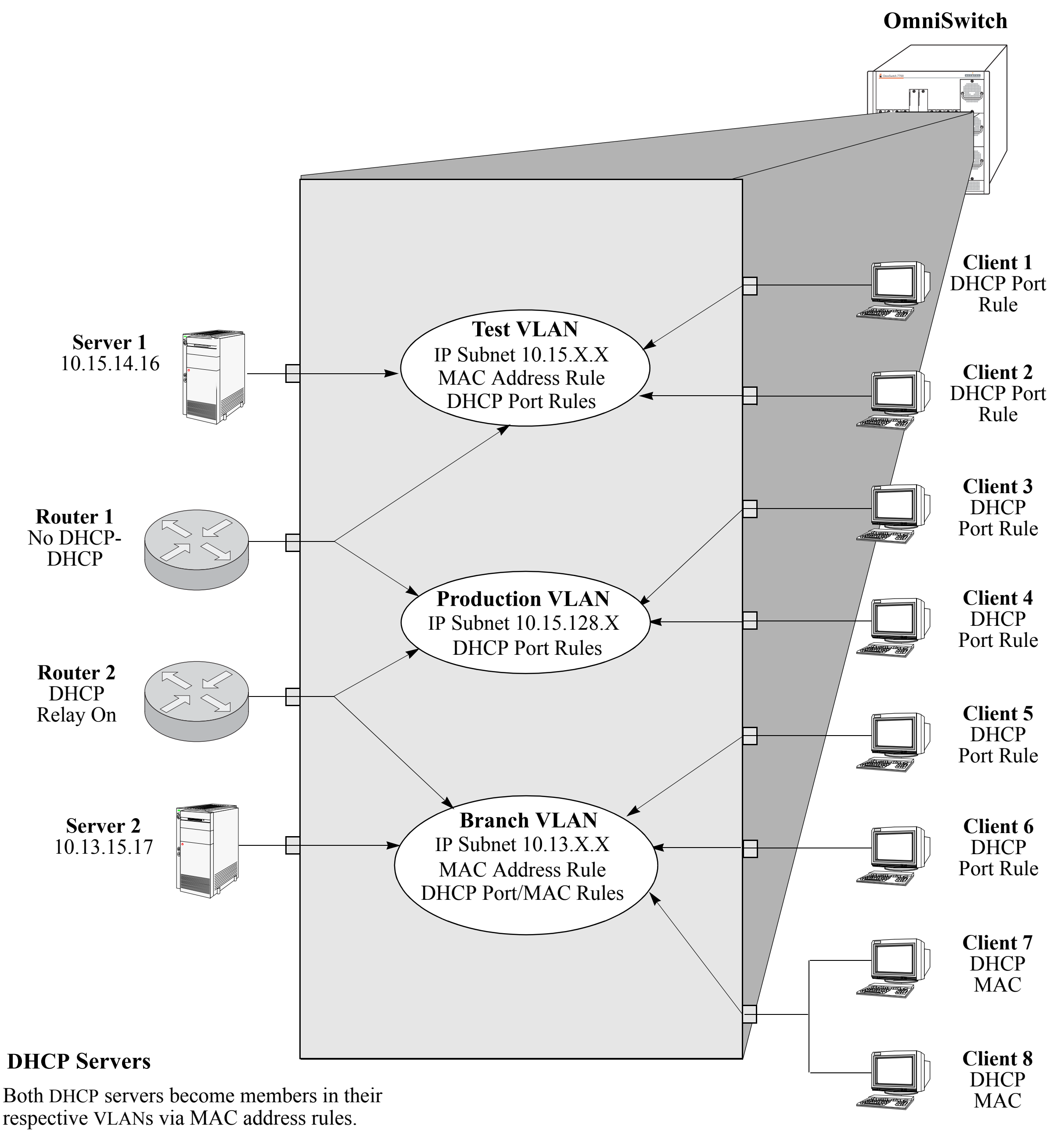
DHCP clients must communicate with a DHCP server at initialization. The most reliable way to ensure this communication is for the server and its associated clients to share the same VLAN. However, if the network configuration does not lend itself to this solution (as the Production VLAN does not in this application example), then the server and clients can communicate through a router with DHCP Relay enabled.

The DHCP servers and clients in this example are either in the same VLAN or are connected through a router with DHCP Relay. All clients in the Test VLAN receive IP addresses from the server in their VLAN (Server 1). Likewise, all clients in the Branch VLAN receive IP addresses from their local server (Server 2). The DHCP clients in the Production VLAN do not have a local DHCP server, so they must rely on the DHCP Relay functionality in external Router 2 to obtain their IP addresses from the DHCP server in the Branch VLAN.

Both DHCP servers are assigned to their VLANs through ~~IP network~~ MAC address rules.

The following table summarizes the VLAN architecture and rules for all devices in this network configuration. The diagram on the following page illustrates this network configuration.

Device	**VLAN Membership**	**Rule Used/Router Role**
DHCP Server 1	Test VLAN	~~IP network address rule=10.15.0.0~~ MAC Address Rule
DHCP Server 2	Branch VLAN	~~IP network address rule=10.13.0.0~~ MAC Address Rule
External Router 1	Test VLAN Production VLAN	Connects Test VLAN to Production VLAN
External Router 2	Production VLAN Branch VLAN	DHCP Relay provides access to DHCP server in Branch VLAN for clients in Production VLAN.
DHCP Client 1	Test VLAN	DHCP Port Rule
DHCP Client 2	Test VLAN	DHCP Port Rule
DHCP Client 3	Production VLAN	DHCP Port Rule
DHCP Client 4	Production VLAN	DHCP Port Rule
DHCP Client 5	Branch VLAN	DHCP Port Rule
DHCP Client 6	Branch VLAN	DHCP Port Rule
DHCP Client 7	Branch VLAN	DHCP MAC Address Rule
DHCP Client 8	Branch VLAN	DHCP MAC Address Rule



DHCP Servers

Both DHCP servers become members in their respective VLANs via MAC address rules.

Routers

Router 1 provides connectivity between the Test VLAN and the Production VLAN. It does not have Bootp functionality enabled so it cannot connect DHCP servers and clients from different VLANs.

Router 2 connects the Production VLAN and the Branch VLAN. With DHCP Relay enabled, this router can provide connectivity between the DHCP server in the Branch VLAN and the DHCP clients in the Production VLAN.

DHCP Clients

Clients 1 to 6 are assigned to their respective VLANs through DHCP port rules. Clients 3 and 4 are not in a VLAN with a DHCP server so they must rely on the server in the Branch VLAN for initial addressing information. Clients 7 and 8 share a port with other devices, so they are assigned to the Branch VLAN via DHCP MAC address rules.

DHCP Port and MAC Rule Application Example

Verifying VLAN Rule Configuration

To display information about VLAN rules configured on the switch, use the show commands listed below:

show vlan rules	Displays a list of rules for one or all VLANs configured on the switch.

For more information about the resulting display from this command, see the *OmniSwitch CLI Reference Guide*. An example of the output for the **show vlan rules** command is also given in "Sample VLAN Rule Configuration" on page 8-3.

9 Using Interswitch Protocols

Alcatel Interswitch Protocols (AIP) are used to discover adjacent switches and retain mobile port information across switches. The following protocol is supported:

- Alcatel Mapping Adjacency Protocol (AMAP), which is used to discover the topology of OmniSwitches and OmniSwitch/Routers (Omni S/R). See "AMAP Overview" on page 9-2.

In This Chapter

This chapter describes the AMAP protocol and how to configure it through the Command Line Interface (CLI). CLI commands are used in the configuration examples; for more details about the syntax of commands, see the *OmniSwitch 7700/7800 CLI Reference Guide*.

Configuration procedures described in this chapter include:

- Activating AMAP on page 9-4.
- Configuring the AMAP discovery timeout interval on page 9-4.
- Configuring the AMAP common timeout interval on page 9-5.

AIP Specifications

Standards	Not applicable at this time. AMAP is Alcatel proprietary protocol.
Maximum number of IP addresses propagated by AMAP	255

AMAP Defaults

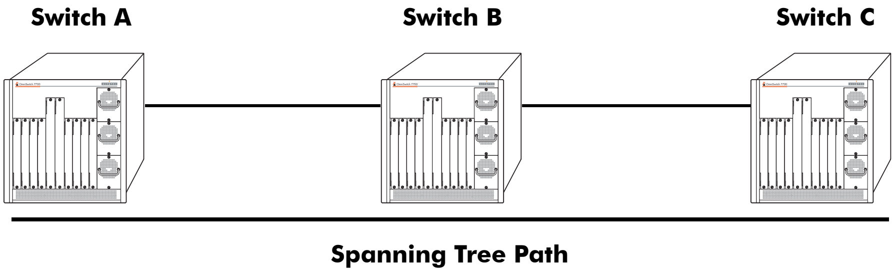
Parameter Description	Command	Default
AMAP status	**amap**	Enabled
Discovery time interval	**amap discovery time**	30 seconds
Common time interval	**amap common time**	300 seconds

AMAP Overview

The Alcatel Mapping Adjacency Protocol (AMAP) is used to discover the topology of OmniSwitches or Omni S/Rs in a particular installation. Using this protocol, each switch determines which OmniSwitches or Omni S/Rs are adjacent to it by sending and responding to Hello update packets. For the purposes of AMAP, adjacent switches are those that:

- have a Spanning Tree path between them
- do not have any switch between them on the Spanning Tree path that has AMAP enabled

In the illustration here, all switches are on the Spanning Tree path. OmniSwitch A and OmniSwitch C have AMAP enabled. OmniSwitch B does not. OmniSwitch A is adjacent to OmniSwitch C and vice versa. If OmniSwitch B enables AMAP, the adjacency changes. OmniSwitch A would be next to B, B would be adjacent to both A and C, and C would be adjacent to B.

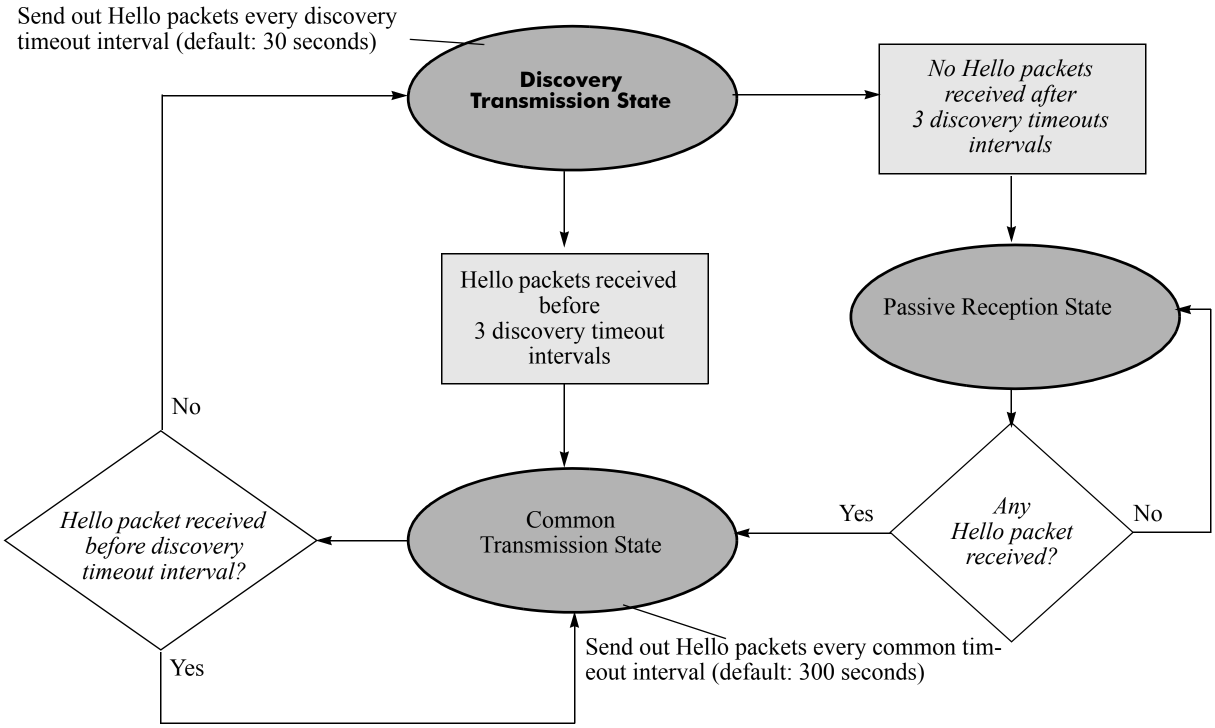


AMAP Transmission States

AMAP switch ports are either in the *discovery transmission state*, *common transmission state*, or *passive reception state*. Ports transition to these states depending on whether or not they receive Hello responses from adjacent switches.

Note. All Hello packet transmissions are sent to a well-known MAC address (0020da:007004).

The transmission states are illustrated here.


Send out Hello packets every discovery timeout interval (default: 30 seconds)
Discovery Transmission State
No Hello packets received after 3 discovery timeouts intervals
Hello packets received before 3 discovery timeout intervals
Passive Reception State
No
Hello packet received before discovery timeout interval?
Common Transmission State
Yes
Any Hello packet received?
No
Yes
Send out Hello packets every common timeout interval (default: 300 seconds)

Discovery Transmission State

When AMAP is active, at startup all active switch ports are in the discovery transmission state. In this state, ports send out Hello packets and wait for Hello responses. Ports send out Hello packets at a configurable interval called the *discovery timeout interval*. This interval is 30 seconds by default. The ports send out Hello packets up to *three* timeouts of this interval trying to discover adjacent switches.

Any switch ports that receive Hello packets send a Hello response and transition to the common transmission state. Any switch ports that do not receive a Hello response before three discovery timeout intervals have expired are placed in the passive reception state.

Common Transmission State

In the common transmission state, ports detect adjacent switch failures or disconnects by sending Hello packets and waiting for Hello responses. Ports send out Hello packets at a configurable interval called the *common timeout interval*. This interval is 300 seconds by default. To avoid synchronization with adjacent switches, the common timeout interval is jittered randomly by plus or minus ten percent.

Ports wait for a Hello response using the discovery timeout interval. If a Hello response is detected within one discovery timeout interval, the port remains in the common transmission state. If a Hello response is not detected within one discovery timeout interval, the port reverts to the discovery transmission state.

Passive Reception State

In the passive reception state, switch ports are in receive-only mode. Hello packets are not sent out from ports in this state and there is no timer on waiting for Hello responses. If the port receives a Hello packet at any time, it enters the common transmission state and transmits a Hello packet in reply.

If a port transitions to the passive reception state, any remote switch entries for that port are deleted.

Common Transmission and Remote Switches

If an AMAP switch is connected to multiple AMAP switches via a hub, the switch sends and receives Hello traffic to and from the remote switches through the same port. If one of the remote switches stops sending Hello packets and other remote switches continue to send Hello packets, the ports in the common transmission state will remain in the common transmission state.

The inactive switch will eventually be aged out of the switch's AMAP database because each remote switch entry has a "last seen" field that is updated when Hello packets are received. The switch checks the "last seen" field at least once every common timeout interval. Switch ports that are no longer "seen" may still retain an entry for up to three common timeout intervals. The slow aging out prevents the port from sending Hello packets right away to the inactive switch and creating additional unnecessary traffic.

Configuring AMAP

AMAP is active by default. In addition to disabling or enabling AMAP, you can view a list of adjacent switches or configure the timeout intervals for Hello packet transmission and reception.

Enabling or Disabling AMAP

To display whether or not AMAP is active or inactive, enter the following command:

```
-> show amap
```

To activate AMAP on the switch, enter the following ommand:

```
-> amap enable
```

To deactivate AMAP on the switch, enter the following command:

```
-> amap disable
```

Configuring the AMAP Discovery Timeout Interval

The discovery timeout interval is used in both the discovery transmission state and the common transmission state to determine how long the port will wait for Hello packets. For ports in the discovery transmission state, this timer is also used as the interval between sending out Hello packets.

Note. Ports in the common transmission state send out Hello packets based on the common timeout interval described later.

The discovery timeout interval is set to 30 seconds by default. To display the current discovery timeout interval, enter the following command:

```
-> show amap
```

To change the discovery timeout interval, use either of these forms of the command with the desired value (any value between 1 and 65535). Note that use of the **time** command keyword is optional. For example:

```
-> amap discovery 60
-> amap discovery time 60
```

Configuring the AMAP Common Timeout Interval

The common timeout interval is used only in the common transmission state to determine the time interval between sending Hello update packets. A switch sends an update for a port just before or after the common timeout interval expires.

Note. Switches avoid synchronization by jittering the common timeout interval plus or minus 10 percent of the configured value. For example, if the default common timeout interval is used (300 seconds), the jitter is plus or minus 30 seconds.

When a Hello packet is received from an adjacent switch before the common timeout interval expires, the switch sends a Hello reply and restarts the common transmission timer.

The common timeout interval is set to 300 seconds by default. To display the current common timeout interval, enter the following command:

```
-> show amap
```

To change the common timeout interval, use either of these forms of the command with the desired value (any value between 1 and 65535). Note that use of the **time** command keyword is optional. For example:

```
-> amap common 600
-> amap common time 600
```

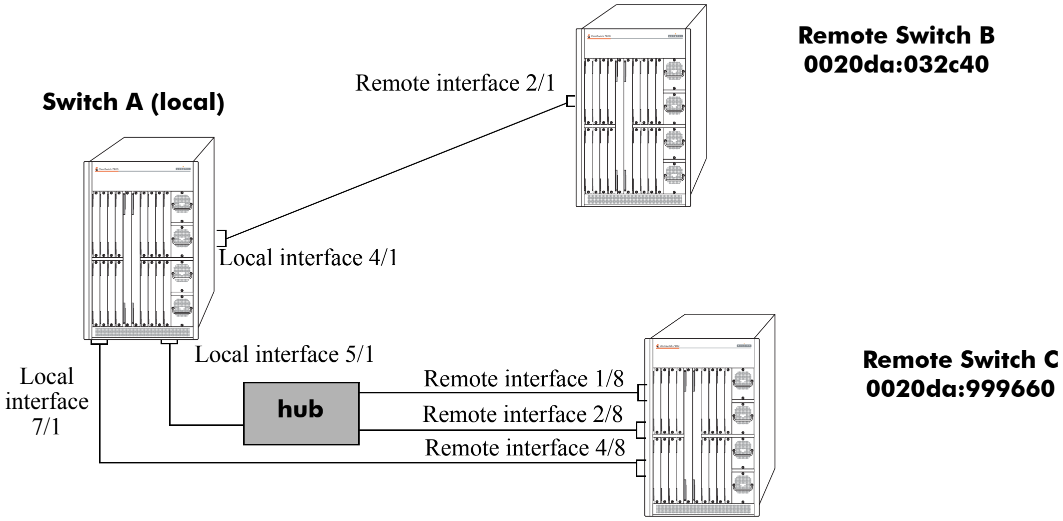
Displaying AMAP Information

Use the **show amap** command to view a list of adjacent switches and their associated MAC addresses, interfaces, VLANs, and IP addresses. For remote switches that stop sending Hello packets and that are connected via a hub, entries may take up to three times the common timeout intervals to age out of this table.

The following example shows three interfaces on a local AMAP switch (4/1, 5/1, 7/1) connected to interfaces on two remote switches. Interface 5/1 is connected to a remote switch through a hub.

```
-> show amap

AMAP:
  Operational Status = enabled,
  Common    Phase Timeout Interval (seconds) = 300,
  Discovery Phase Timeout Interval (seconds) = 30

Remote Host 'Switch B' On Port 4/1 Vlan 1:
Remote Device          = OS6800,
Remote Base MAC        = 00:20:da:03:2c:40,
Remote Interface       = 2/1,
Remote VLAN            = 1,
Number of Remote IP Address(es) Configured = 4,
Remote IP(s) =
18.1.1.1
27.0.0.2
192.168.10.1
192.206.184.40

Remote Host 'Switch C' On Port 5/1 Vlan 7:
Remote Device          = OS6800,
Remote Base MAC        = 00:20:da:99:96:60,
Remote Interface       = 1/8,
Remote Vlan            = 7,
Number of Remote IP Address(es) Configured = 1,
Remote IP(s) =
192.206.184.20

Remote Host 'Switch C' On Port 5/1 Vlan 7:
Remote Device          = OS6800,
Remote Base MAC        = 00:20:da:99:96:60,
Remote Interface       = 2/8,
Remote Vlan            = 255,
Number of Remote IP Address(es) Configured = 1,
Remote IP(s) =
192.206.185.30

Remote Host 'Switch C' On Port 7/1 Vlan 455:
Remote Device          = OS6800,
Remote Base MAC        = 00:20:da:99:96:60,
Remote Interface       = 4/8,
Remote Vlan            = 455,
Number of Remote IP Address(es) Configured = 3,
Remote IP(s) =
192.206.183.10
192.206.184.20
192.206.185.30
```

A simplified visual illustration of these connections is shown here for example purposes only:



See the *OmniSwitch CLI Reference Guide* for information about the **show amap** command.

10 Configuring 802.1Q

802.1Q is the IEEE standard for segmenting networks into VLANs. 802.1Q segmentation is done by adding a specific tag to a packet.

In this Chapter

This chapter describes the basic components of 802.1Q VLANs and how to configure them through the Command Line Interface (CLI). CLI commands are used in the configuration examples; for more details about the syntax of commands, see “802.1Q Commands” in the *OmniSwitch CLI Reference Guide*.

Configuration procedures described in this chapter include:

- Setting up an 802.1Q VLAN for a specific port. See “Enabling Tagging on a Port” on page 10-5.
- Setting up an 802.1Q VLAN for an link aggregation group. See “Enabling Tagging with Link Aggregation” on page 10-6.
- Configuring 802.1Q VLAN parameters. See “Configuring the Frame Type” on page 10-7 and “Forcing a Switch Internal Tag” on page 10-7.

For information on creating and managing VLANs, see Chapter 5, “Configuring VLANs.”

For information on creating and managing link aggregation groups, see Chapter 11, “Configuring Static Link Aggregation” and Chapter 12, “Configuring Dynamic Link Aggregation.”

802.1Q Specifications

IEEE Specification	*Draft Standard P802.1Q/D11 IEEE Standards for Local And Metropolitan Area Network: Virtual Bridged Local Area Networks, July 30, 1998*
Maximum Number of Tagged VLANs per Port	4093
Maximum Number of Untagged VLANs per Port	One untagged VLAN per port.
Maximum Number of VLAN Port Associations	32768

Note. Up to 4093 VLANs can be assigned to a tagged port or link aggregation group. However, each assignment counts as a single VLAN port association. Once the maximum number of VLAN port associations is reached, no more VLANs can be assigned to ports. For more information, see the chapter titled Chapter 7, "Assigning Ports to VLANs."

802.1Q Defaults Table

The following table shows the default settings of the configurable 802.1Q parameters.

802.1Q Defaults

Parameter Description	Command	Default Value/Comments
What type of frames accepted	**vlan 802.1q frame type**	Both tagged and untagged frames are accepted
Which tag is enforced	**vlan 802.1q force tag internal**	On

802.1Q Overview

Alcatel’s 802.1Q is an IEEE standard for sending frames through the network tagged with VLAN identification. This chapter details procedures for configuring and monitoring 802.1Q tagging on a single port in a switch or a link aggregation group in a switch.

802.1Q tagging is the IEEE version of VLANs. It is a method for segregating areas of a network into distinct VLANs. By attaching a label, or tag, to a packet, the packet can be identified as being from a specific area or identified as being destined for a specific area.

When enabling a tagged port, you will also need to specify whether only 802.1Q tagged traffic is allowed on the port, or whether the port accepts both tagged and untagged traffic.

“Tagged” refers to four bytes of reserved space in the header of the packet. The four bytes of “tagging” are broken down as follows: the first two bytes indicate whether the packet is an 802.1Q packet, and the next two bytes carry the VLAN Identification (VID) and priority.

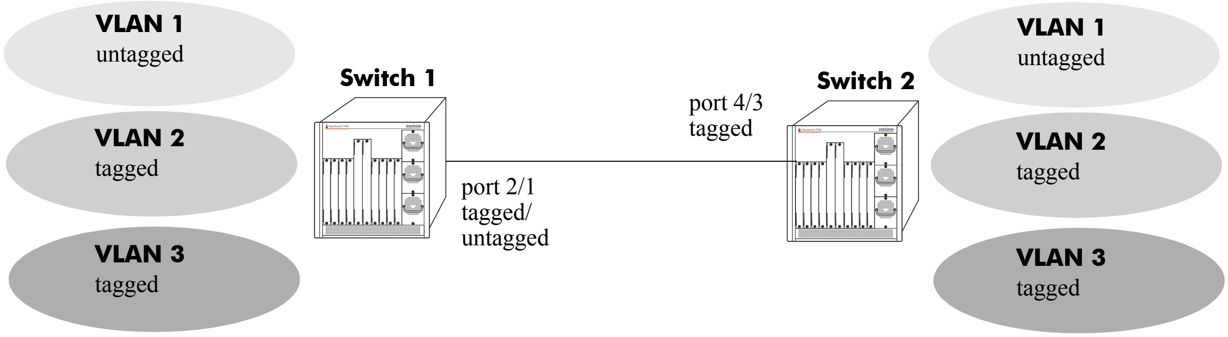
On the ingress side, packets are classified in a VLAN. After classifying a packet, the switch adds an 802.1Q header to the packet. Egress processing of packets is done by the switch hardware. Packets have an 802.1Q tag which may be stripped off based on 802.1Q tagging/stripping rules.

If a port is configured to be a tagged port, then all the untagged traffic (including priority tagged, or VLAN 0 traffic) received on the port will be dropped. You do not need to reboot the switch after changing the configuration parameters.

Note. Priority tagged traffic, or traffic from VLAN 0, is used for Quality of Service (QoS) functionality. 802.1Q views priority tagged traffic as untagged traffic.

Ports that are configured to accept 802.1Q tagged traffic cannot be mobile ports. They are statically assigned to VLANs.

The following diagram illustrates a simple network using tagged and untagged traffic:



Tagged and Untagged Traffic Network

Switch 1 and 2 have three VLANs, one for untagged traffic and two for tagged traffic. The ports connecting Switch 1 and 2 are configured in such a manner that Port 4/3 will only accept tagged traffic, while Port 2/1 will accept both tagged and untagged traffic.

The port can only be assigned to one untagged VLAN (in every case, this will be the default VLAN). In the example above the default VLAN is VLAN 1. The port can be assigned to as many 802.1Q VLANs as necessary, up to 4093 per port or 32768 VLAN port associations.

For the purposes of Quality of Service (QoS), 802.1Q ports are always considered to be *trusted* ports. For more information on QoS and trusted ports, see Chapter 24, "Configuring QoS."

Alcatel's 802.1Q tagging is done at wire speed, providing high-performance throughput of tagged frames.

The procedures below use CLI commands that are thoroughly described in "802.1Q Commands" of the *OmniSwitch CLI Reference Guide.*

Configuring an 802.1Q VLAN

The following sections detail procedures for creating 802.1Q VLANs and assigning ports to 802.1Q VLANs.

Enabling Tagging on a Port

To set a port to be a tagged port, you must specify a VLAN identification (VID) number and a port number. You may also optionally assign a text identification.

For example, to configure port 4 on slot 3 to be a tagged port, enter the following command at the CLI prompt:

```
-> vlan 5 802.1q 3/4
```

Tagging would now be enabled on port 3/4, with a VID of 5.

To add tagging to a port, and label it with a text name, you would enter the text identification following the slot and port number. For example, to enable tagging on port 4 of slot 3 with a text name of **port tag**, enter the command in the following manner:

```
-> vlan 5 802.1q 3/4 "port tag"
```

The tagged port would now also be labeled **port tag**. Note that you must use quotes around the text description.

The VLAN used to handle traffic on the tagged port must be created prior to using the **vlan 802.1q** command. Creating a VLAN is described in Chapter 5, "Configuring VLANs."

For more specific information, see the **vlan 802.1q** command section in the *OmniSwitch CLI Reference Guide*.

Enabling Tagging with Link Aggregation

To enable tagging on link aggregation groups, enter the link aggregation group identification number in place of the slot and port number, as shown:

```
-> vlan 5 802.1q 8
```

(For further information on creating link aggregation groups, see Chapter 11, “Configuring Static Link Aggregation,” or Chapter 12, “Configuring Dynamic Link Aggregation.”)

To add tagging to a port or link aggregation group, and label it with a text name, you would enter the text identification following the slot and port number or link aggregation group identification number. For example, to enable tagging on link aggregation group 8 with a text name of **agg port tag**, enter the command in the following manner:

```
-> vlan 5 802.1q 8 “agg port tag”
```

The tagged port would now also be labeled **agg port tag**. Note that you must use quotes around the text description.

To remove 802.1Q tagging from a selected port, use the same command as above with a **no** keyword added, as shown:

```
-> vlan 5 no 802.1q 8
```

Note. The link aggregation group must be created first before it can be set to use 802.1Q tagging

For more specific information, see the **vlan 802.1q** command section in the *OmniSwitch CLI Reference Guide*.

Configuring the Frame Type

Once a port has been set to receive and send tagged frames, it will be able to receive or send tagged or untagged traffic. Tagged traffic will be subject to 802.1Q rules, while untagged traffic will behave as directed by normal switch operation. (Setting up rules for non-802.1Q traffic is defined in Chapter 5, "Configuring VLANs.") A port can also be configured to accept only tagged frames.

To configure a port to only accept tagged frames, enter the **frame type** command at the CLI prompt:

```
-> vlan 802.1q 3/4 frame type tagged
```

To configure a port back to accepting both tagged and untagged traffic, use the same command with the **all** keyword, as shown:

```
-> vlan 802.1q 3/4 frame type all
```

Note. If you configure a port to accept only VLAN-tagged frames, then any frames received on this port that do not carry a VLAN identification (i.e., untagged frames or priority-tagged frames) will be discarded by the ingress rules for this port. Frames that are not discarded by this ingress rule are classified and processed according to the ingress rules for this port.

When a port is set to support both tagged and untagged traffic, multiple VLANs for 802.1Q traffic can be added to the port, but only one VLAN can be used to support untagged traffic. The untagged traffic VLAN will always be the port's default VLAN.

Note. You cannot configure an link aggregation group to accept only tagged frames.

For more specific information, see the **vlan 802.1q frame type** command section in the *OmniSwitch CLI Reference Guide*.

Forcing a Switch Internal Tag

If a tagged packet comes in on an untagged or mobile port (i.e., the **vlan 802.1q** command has not been used), then it can be classified in a VLAN other than the VLAN to which it currently belongs. If this classified VLAN (i.e., different than the packet tag) is now tagged on the egress side, then there are two possible options. One option is to carry the original tag of the packet. The other option is to replace the original tag with the classified VLAN. If **force tag internal** is **on**, then the tag is replaced with the classified VLAN. If the force tag internal is **off**, then the tag is not replaced with the classified VLAN as the tag.

You *must* enable 802.1 tagging, described in "Enabling Tagging on a Port" on page 10-5 command, before you can use this command.

Force tag internal is **on** by default. Turning off the feature is done as shown:

```
-> vlan 802.1q 3/4 force tag internal off
```

To activate the internal forced tag feature for a port, enter the **force tag internal** command as shown:

```
-> vlan 802.1q 3/4 force tag internal on
```

For more specific information, see the **vlan 802.1q force tag internal** command section in the *OmniSwitch CLI Reference Guide*.

Show 802.1Q Information

After configuring a port or link aggregation group to be a tagged port, you can view the settings by using the **show 802.1q** command, as demonstrated:

```
-> show 802.1q 3/4

Acceptable Frame Type   :      Any Frame Type
Force Tag Internal      :                off

Tagged VLANS     Internal Description
-------------+-------------------------------------------------+
          2      TAG PORT 3/4 VLAN 2



-> show 802.1q 2

Tagged VLANS     Internal Description
-------------+-------------------------------------------------+
          3      TAG AGGREGATE 2 VLAN 3
```

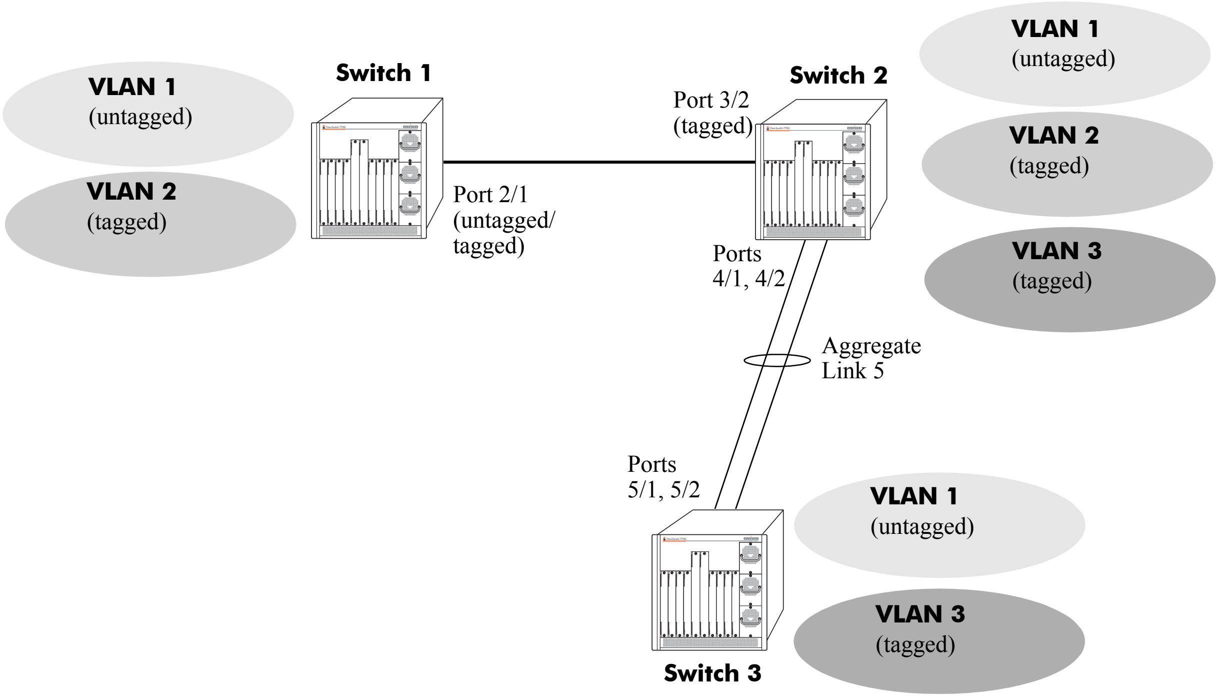
To display all VLANs, enter the following command:

```
-> show vlan port
```

Application Example

In this section the steps to create 802.1Q connections between switches are shown.

The following diagram shows a simple network employing 802.1Q on both regular ports and link aggregation groups.



The following sections show how to create the network illustrated above.

Connecting Switch 1 and Switch 2 Using 802.1Q

The following steps apply to Switch 1. They will attach port 2/1 to VLAN 2, and set the port to accept 802.1Q tagged traffic and untagged traffic.

1 Create VLAN 2 by entering **vlan 2** as shown below (VLAN 1 is the default VLAN for the switch):

```
-> vlan 2
```

2 Set port 2/1 as a tagged port and assign it to VLAN 2 by entering the following:

```
-> vlan 2 802.1q 2/1
```

3 Check the configuration using the **show 802.1q** command, as follows:

```
-> show 802.1q 2/1

Acceptable Frame Type    :      Any Frame Type
Force Tag Internal       :                  on

Tagged VLANS      Internal Description
-------------+-------------------------------------------------+
           2      TAG PORT 2/1 VLAN 2
```

The following steps apply to Switch 2. They will attach port 3/2 to VLAN 2, and set the port to accept 802.1Q tagged traffic only:

1 Create VLAN 2 by entering **vlan 2** as shown below (VLAN 1 is the default VLAN for the switch):

```
-> vlan 2
```

2 Set port 3/2 as a tagged port and assign it to VLAN 2 by entering the following:

```
-> vlan 2 802.1q 3/2
```

3 Set port 3/2 to accept only tagged traffic by entering the following:

```
-> vlan 802.1q 3/2 frame type tagged
```

4 Check the configuration using the **show 802.1q** command, as follows:

```
-> show 802.1q 3/2

Acceptable Frame Type   :        tagged only
Force Tag Internal      :                 on

Tagged VLANS      Internal Description
-------------+-------------------------------------------------+
           2      TAG PORT 3/2 VLAN 2
```

Connecting Switch 2 and Switch 3 Using 802.1Q

The following steps apply to Switch 2. They will attach ports 4/1 and 4/2 as link aggregation group 5 to VLAN 3.

1 Configure static aggregate VLAN 5 by entering the following:

```
-> static linkagg 5 size 2
```

2 Assign ports 4/1 and 4/2 to static aggregate VLAN 5 by entering the following two commands:

```
-> static agg 4/1 agg num 5
-> static agg 4/2 agg num 5
```

3 Create VLAN 3 by entering the following:

```
-> vlan 3
```

4 Configure 802.1Q tagging with a tagging ID of 3 on link aggregation group 5 (on VLAN 3) by entering **vlan 3 802.1q 5** as shown below:

```
-> vlan 3 802.1q 5
```

5 Check the configuration using the **show 802.1q** command, as follows:

```
-> show 802.1q 5

Tagged VLANS      Internal Description
-------------+-------------------------------------------------+
           3      TAG AGGREGATE 5 VLAN 3
```

The following steps apply to Switch 3. They will attach ports 5/1 and 5/2 as link aggregation group 5 to VLAN 3.

1 Configure static link aggregation group 5 by entering the following:

```
-> static linkagg 5 size 2
```

2 Assign ports 5/1 and 5/2 to static link aggregation group 5 by entering the following two commands:

```
-> static agg 5/1 agg num 5
-> static agg 5/2 agg num 5
```

3 Create VLAN 3 by entering the following:

```
-> vlan 3
```

4 Configure 802.1Q tagging with a tagging ID of 3 on static link aggregation group 5 (on VLAN 3) by entering the following:

```
-> vlan 3 802.1q 5
```

5 Check the configuration using the **show 802.1q** command, as follows:

```
-> show 802.1q 5

Tagged VLANS      Internal Description
-------------+-------------------------------------------------+
           3      TAG AGGREGATE 5 VLAN 3
```

Verifying 802.1Q Configuration

To display information about the ports configured to handle tagging, use the following show command:

show 802.1q	Displays 802.1Q tagging information for a single port or a link aggregation group.

For more information about the resulting display, see Chapter 1, “802.1Q Commands,” in the *OmniSwitch CLI Reference Guide*.

11 Configuring Static Link Aggregation

Alcatel's static link aggregation software allows you to combine several physical links into one large virtual link known as a link aggregation *group*. Using link aggregation can provide the following benefits:

- **Scalability**. You can configure up to 32 link aggregation groups on an OmniSwitch 7700/7800 or 16 link aggregation groups on an OmniSwitch 8800 that can consist of 2, 4, 8, or 16 10 Mbps, 100 Mbps, or 1 Gbps Ethernet links.
- **Reliability**. If one of the physical links in a link aggregate group goes down (unless it is the last one) the link aggregate group can still operate.
- **Ease of Migration**. Link aggregation can ease the transition from a 100 Mbps Ethernet backbones to Gigabit Ethernet backbones.

Note. This chapter only covers dynamic link aggregation for OmniSwitch 7700, 7800, and 8800 switches. Please refer to the *OmniSwitch 6600 Family Network Configuration Guide* for information on configuring dynamic link aggregation on OmniSwitch 6600 Family switches and to the *OmniSwitch 6800 Series Network Configuration Guide* for information on configuring dynamic link aggregation on OmniSwitch 6800 Series switches. These switches use different procedures and have many different operating ranges.

In This Chapter

This chapter describes the basic components of static link aggregation and how to configure them through the Command Line Interface (CLI). CLI commands are used in the configuration examples; for more details about the syntax of commands, see the *OmniSwitch CLI Reference Guide*.

Configuration procedures described in this chapter include:

- Configuring static link aggregation groups on page 11-7.
- Adding and deleting ports from a static aggregate group on page 11-9.
- Modifying static link aggregation default values on page 11-11.

Note. You can also configure and monitor static link aggregation with WebView, Alcatel's embedded web-based device management application. WebView is an interactive and easy-to-use GUI that can be launched from OmniVista or a web browser. Please refer to WebView's online documentation for more information on configuring and monitoring static link aggregation with WebView.

Static Link Aggregation Specifications

The table below lists specifications for static groups.

Maximum number of link aggregation groups	32 (OmniSwitch 7700/7800) 16 (OmniSwitch 8800)
Number of links per group supported	2, 4, 8, or 16
Range for optional group name	1 to 255 characters
CLI Command Prefix Recognition	All static link aggregation configuration commands support prefix recognition. (Static link aggregation show commands do not support prefix recognition.) See the "Using the CLI" chapter in the *OmniSwitch 7700/7800/8800 Switch Management Guide* for more information.

Static Link Aggregation Default Values

The table below lists default values and the commands to modify them for static aggregate groups.

Parameter Description	Command	Default Value/Comments
Administrative State	**static linkagg admin state**	enabled
Group Name	**static linkagg name**	No name configured
Link aggregation optimization on First-generation NI modules	**linkagg slot optimization**	disabled
Single or multiple link aggregation groups on First-generation NI modules	**linkagg slot single** **linkagg slot multiple**	single

Quick Steps for Configuring Static Link Aggregation

Follow the steps below for a quick tutorial on configuring a static aggregate link between two switches. Additional information on how to configure each command is given in the subsections that follow.

1 Create the static aggregate link on the local switch with the **static linkagg size** command. For example:

```
-> static linkagg 1 size 4
```

2 Assign all the necessary ports to the static link aggregation group on the local switch with the **static agg agg num** command. For example:

```
-> static agg 4/1 agg num 1
-> static agg 4/7 agg num 1
-> static agg 5/1 agg num 1
-> static agg 6/2 agg num 1
```

3 Create a VLAN for this static link aggregate group with the **vlan** command. For example:

```
-> vlan 10 port default 1
```

4 Create the equivalent static aggregate link on the remote switch with the **static linkagg size** command. For example:

```
-> static linkagg 1 size 4
```

5 Assign all the necessary ports to the static link aggregation group on the remote switch with the **static agg agg num** command. For example:

```
-> static agg 7/1 agg num 1
-> static agg 7/7 agg num 1
-> static agg 8/1 agg num 1
-> static agg 9/2 agg num 1
```

6 Create a VLAN for this static link aggregate group with the **vlan** command. For example:

```
-> vlan 10 port default 1
```

Note. *Optional*. You can verify your static link aggregation settings with the **show linkagg** command. For example:

```
-> show linkagg 1
Static Aggregate
  SNMP Id                    : 40000001,
  Aggregate Number           : 1,
  SNMP Descriptor            : Omnichannel Aggregate Number 1 ref 40000001 size 4,
  Name                       : ,
  Admin State                : ENABLED,
  Operational State          : UP,
  Aggregate Size             : 4,
  Number of Selected Ports : 4,
  Number of Reserved Ports : 4,
  Number of Attached Ports : 4,
  Primary Port               : 4/1
```

You can also use the **show linkagg port** port command to display information on specific ports. See "Displaying Static Link Aggregation Configuration and Statistics" on page 11-15 for more information on **show** commands.

An example of what these commands look like entered sequentially on the command line on the local switch:

```
-> static linkagg 1 size 4
-> static agg 4/1 agg num 1
-> static agg 4/7 agg num 1
-> static agg 5/1 agg num 1
-> static agg 6/2 agg num 1
-> vlan 10 port default 1
```

An example of what these commands look like entered sequentially on the command line on the remote switch:

```
-> static linkagg 1 size 4
-> static agg 7/1 agg num 1
-> static agg 7/7 agg num 1
-> static agg 8/1 agg num 1
-> static agg 9/2 agg num 1
-> vlan 10 port default 1
```

Static Link Aggregation Overview

Link aggregation allows you to combine 2, 4, 8, or 16 physical connections into large virtual connections known as link aggregation *groups*. On an OmniSwitch 7700/7800 switch you can create up to 32 link aggregation (both dynamic and static) groups. On an OmniSwitch 8800 switch you can create up to 16 link aggregation (both dynamic and static) groups. You can create Virtual LANs (VLANs), configure Quality of Service (QoS) conditions, 802.1Q framing, and other networking features on link aggregation groups because the switch's software treats these virtual links just like physical links. (See "Relationship to Other Features" on page 11-6 for more information on how link aggregation interacts with other software features.)

Load balancing for Layer 2 non IP packets is on a MAC address basis and for IP packets the balancing algorithm uses IP address as well. Ports must be the same speed within the same link aggregate group.

Alcatel's link aggregation software allows you to configure the following two different types of link aggregation groups:

- Static link aggregate groups
- Dynamic link aggregate groups

This chapter describes static link aggregation. For information on dynamic link aggregation, please refer to Chapter 12, "Configuring Dynamic Link Aggregation."

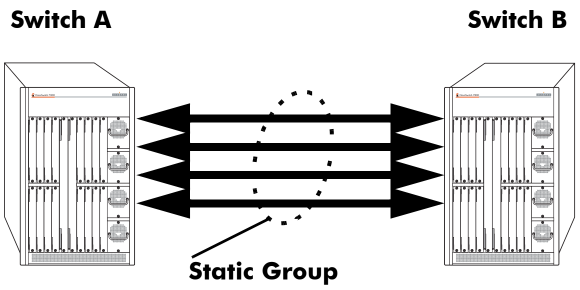
Static Link Aggregation Operation

Static link aggregate groups are virtual links between two switches consisting of 2, 4, 8, or 16 10 Mbps, 100 Mbps, or 1 Gbps fixed physical links. These links can be on different Network Interfaces (NIs) and do not need to be assigned sequentially. Up to 32 static and or dynamic link aggregation groups can be configured on an OmniSwitch 7700/7800 switch and up to 16 static and or dynamic link aggregation groups can be configured on an OmniSwitch 8800 switch.

Static aggregate groups can be created between two OmniSwitch 7700/7800/8800 switches or between an OmniSwitch 7700/7800/8800 and an early-generation Alcatel switch such as an Omni Switch/Router. However, static aggregate groups cannot be created between OmniSwitch 7700/7800/8800 switches and some switches from other vendors.

The figure below shows a static aggregate group that has been configured between Switch A and Switch B. The static aggregate group links four ports on a single OS7-ENI-FM12 on Switch A to two ports on one OS7-ENI-FM12 and two ports on another OS7-ENI-FM12 on Switch B. The network administrator has created a separate VLAN for this group so users can use this high speed link.

Switch software treats the
static aggregate groups as
one large virtual link.



Example of a Static Link Aggregate Group Network

See "Configuring Static Link Aggregation Groups" on page 11-7 for information on using Command Line Interface (CLI) commands to configure static aggregate groups and see "Displaying Static Link Aggregation Configuration and Statistics" on page 11-15 for information on using CLI to monitor static aggregate groups.

Relationship to Other Features

Link aggregation groups are supported by other switch software features. The following features have CLI commands or command parameters that support link aggregation:

- **VLANs**. For more information on VLANs see Chapter 5, "Configuring VLANs."
- **802.1Q**. For more information on configuring and monitoring 802.1Q see Chapter 10, "Configuring 802.1Q."
- **Spanning Tree**. For more information on Spanning Tree see Chapter 6, "Configuring Spanning Tree Parameters."

Note. See "Application Example" on page 11-13 for tutorials on using link aggregation with other features.

Configuring Static Link Aggregation Groups

This section describes how to use Alcatel's Command Line Interface (CLI) commands to configure static link aggregate groups. See "Configuring Mandatory Static Link Aggregate Parameters" on page 11-7 for more information.

Note. See "Quick Steps for Configuring Static Link Aggregation" on page 11-3 for a brief tutorial on configuring these mandatory parameters.

Alcatel's link aggregation software is preconfigured with the default values for static aggregate groups as shown in the table in "Static Link Aggregation Default Values" on page 11-2. If you need to modify any of these parameters, please see "Modifying Static Aggregation Group Parameters" on page 11-11 for more information.

Note. See the "Link Aggregation Commands" chapter in the *OmniSwitch CLI Reference Guide* for complete documentation of CLI commands for link aggregation.

Configuring Mandatory Static Link Aggregate Parameters

When configuring static link aggregates on a switch you must perform the following steps:

Note. Depending on the hardware version of your Network Interface (NI) modules, you may need to optimize those modules for link aggregation. See "Optimizing NI Modules for Link Aggregation" on page 11-12 for more information. In addition, if you optimize an NI for link aggregation only one link aggregation group can be configured on this module. To configure additional link aggregation groups see "Configuring Multiple Link Aggregation Groups on NI Modules" on page 11-12 for more information.

1 Create the Static Aggregate Group on the Local and Remote Switches. To create a static aggregate group use the **static linkagg size** command, which is described in "Creating and Deleting a Static Link Aggregate Group" on page 11-8.

2 Assign Ports on the Local and Remote Switches to the Static Aggregate Group. To assign ports to the static aggregate group you use the **static agg agg num** command, which is described in "Adding and Deleting Ports in a Static Aggregate Group" on page 11-9.

Note. Depending on the needs of your network you may need to configure additional parameters. Commands to configure optional static aggregate parameters are described in "Modifying Static Aggregation Group Parameters" on page 11-11.

Creating and Deleting a Static Link Aggregate Group

The following subsections describe how to create and delete static link aggregate groups with the **static linkagg size** command.

Creating a Static Aggregate Group

To create a static aggregate group on a switch by entering **static linkagg** followed by the user-specified aggregate number (which can be 0 through 31 on an OmniSwitch 7700/7800 switch and 0 through 15 on an OmniSwitch 8800 switch), **size**, and the number of links in the static aggregate group (which can be 2, 4, 8, or 16).

For example, to create static aggregate group 5 that consists of eight (8) links on a switch you would enter:

```
-> static linkagg 5 size 8
```

You can create up to 32 aggregation groups (both static and dynamic) on an OmniSwitch 7700/7800 switch and up to 16 aggregation groups (both static and dynamic) on an OmniSwitch 8800 switch. However, due to the number of hardware queues available for link aggregation, the maximum number of link aggregate groups decreases as the aggregate group size (i.e., number of links) increases. See the table below for more information on OmniSwitch 7700/7800.

Number of Links (Aggregate Size) on OmniSwitch 7700/7800	Maximum Number of Aggregate Groups on a Switch
2	32 (range 0–31)
4	29 (range 0–28)
8	14 (range 0–13)
16	7 (range 0–6)

And see the table below for OmniSwitch 8800:

Number of Links (Aggregate Size) on OmniSwitch 8800	Maximum Number of Aggregate Groups on a Switch
2	16 (range 0–15)
4	16 (range 0–15)
8	14 (range 0–13)
16	7 (range 0–6)

Note. The number of links assigned to a static aggregate group should always be close to the number of physical links that you plan to use. For example, if you are planning to use 4 physical links you should create a group with a size of 4 and not 8 or 16.

As an option you can also specify a name and/or the administrative status of the group by entering **static linkagg** followed by the user-specified aggregate number, **size**, the number of links in the static aggregate group, **name**, the optional name (which can be up to 255 characters long), **admin state**, and either **enable** or **disable** (the default is **enable**).

For example, to create static aggregate group 5 called "static1" consisting of eight (8) links that is administratively disabled enter:

```
-> static linkagg 5 size 8 name static1 admin state disable
```

Note. If you want to specify spaces within a name for a static aggregate group the name must be specified within quotes (e.g., "Static Aggregate Group 5").

Deleting a Static Aggregate Group

To delete a static aggregation group from a switch use the **no** form of the **static linkagg size** command by entering **no static linkagg** followed by the number that identifies the group. For example, to remove static aggregate group 5 from a switch's configuration you would enter:

```
-> no static linkagg 5
```

Note. You must delete any attached ports with the **static agg agg num** command before you can delete a static link aggregate group.

Adding and Deleting Ports in a Static Aggregate Group

The following subsections describe how to add and delete ports in a static aggregate group with the **static agg agg num** command.

Note. Link Aggregation is not supported on OS8-10GNI-UR1 modules.

Adding Ports to a Static Aggregate Group

The number of ports assigned in a static aggregate group can be less than or equal to the maximum size you specified in the **static linkagg size** command. To assign a port to a static aggregate group you use the **static agg agg num** command by entering **static agg** followed by the slot number, a slash (/), the port number, **agg num**, and the number of the static aggregate group.

For example, to assign port 1 in slot 5 to static aggregate group 10 you would enter:

```
-> static agg 5/1 agg num 10
```

Note. A port may belong to only one aggregate group. In addition, mobile ports cannot be aggregated. See Chapter 7, "Assigning Ports to VLANs," for more information on mobile ports.

As an option, you can use the **ethernet**, **fastethernet**, and **gigaethernet** keywords before the slot and port number to document the interface type to make the command look consistent with early-generation Alcatel CLI syntax. For example, to assign port 1 in slot 5 to static aggregate group 10 and document that port 1 in slot 5 is a Fast Ethernet port you would enter:

```
-> static fastethernet agg 5/1 agg num 10
```

Note. The the **ethernet**, **fastethernet**, and **gigaethernet** keywords do not modify a port's configuration. See Chapter 1, "Configuring Ethernet Ports," for information on configuring Ethernet ports.

Removing a Port from a Static Aggregate Group

To remove a port from a static aggregate group you use the **no** form of the **static agg agg num** command by entering **static agg no** followed by the slot number, a slash (/), and the port number. For example, to remove port 1 in slot 5 from a static aggregate group you would enter:

```
-> static agg no 5/1
```

Modifying Static Aggregation Group Parameters

This section describes how to modify the following static aggregate group parameters:

- Static aggregate group name (see "Modifying the Static Aggregate Group Name" on page 11-11)
- Static aggregate group administrative state (see "Modifying the Static Aggregate Group Administrative State" on page 11-11)

Modifying the Static Aggregate Group Name

The following subsections describe how to modify the name of the static aggregate group with the **static linkagg name** command.

Creating a Static Aggregate Group Name

To create a name for a static aggregate group by entering **static linkagg** followed by the number of the static aggregate group, **name**, and the user-specified name of the group, which can be up to 255 characters long. For example, to configure static aggregate group 4 with the name "Finance" you would enter:

```
-> static linkagg 4 name Finance
```

Note. If you want to specify spaces within a name for a static aggregate group the name must be specified within quotes (e.g., "Static Aggregate Group 4").

Deleting a Static Aggregate Group Name

To remove a name from a static aggregate group you use the **no** form of the **static linkagg name** command by entering **static linkagg** followed by the number of the static aggregate group and **no name**. For example, to remove any user-specified name from static aggregate group 4 you would enter:

```
-> static linkagg 4 no name
```

Modifying the Static Aggregate Group Administrative State

By default, the administrative state for a static aggregate group is enabled. The following subsections describe how to enable and disable the administrative state with the **static linkagg admin state** command.

Enabling the Static Aggregate Group Administrative State

To enable a static aggregate group by entering **static linkagg** followed by the number of the group and **admin state enable**. For example, to enable static aggregate group 1 you would enter:

```
-> static linkagg 1 admin state enable
```

Disabling the Static Aggregate Group Administrative State

To disable a static aggregate group by entering **static linkagg** followed by the number of the group and **admin state disable**. For example, to disable static aggregate group 1 you would enter:

```
-> static linkagg 1 admin state disable
```

Optimizing NI Modules for Link Aggregation

By default, first-generation Network Interface (NI) modules are not optimized for link aggregation. The table below shows which NI modules are first-generation modules and are not optimized for link aggregation. Use the **show ni**, **show module**, or **show ni** command to display the part number of the NI module. Second-generation NI modules are distinguished from first-generation NI modules by "ENI2" or "GNI2" in the part number. (First-generation modules have ENI, GNI, or 10GNI in their part numbers instead.) If the NI is a second-generation module you do not need to optimize it.

To modify the optimization status of an NI module use the **linkagg slot optimization** command. To use this command enter **linkagg slot** followed by the slot number of the NI module and either **enable** or **disable**.

For example, to enable link aggregation optimization on an NI module is Slot 5 enter:

```
-> linkagg slot 5 optimization enable
```

When a port is a member of an aggregate group and optimization is enabled on this NI, all bridged traffic sent from any other port (not part of the aggregate group) on the same switching ASIC to the aggregate will be dropped. In this case, traffic needs to be routed between that port and the aggregate group.

In a chassis with both first-generation and second-generation NI modules you must configure static link aggregation on all of the first-generation NI modules before you *must* configure static link aggregation on all of these NI modules before you configure it on any of the second-generation NI modules. In addition, hot insertion or hot swapping of a first-generation NI module into a chassis that has only second-generation NI modules can cause configuration problems.

Configuring Multiple Link Aggregation Groups on NI Modules

By default, only a single link aggregation group can be configured on a first-generation Network Interface (NI) module if the NI module has been optimized for link aggregation with the **linkagg slot optimization** command. To configure multiple link aggregation groups on these NI modules use the **linkagg slot multiple** command. To use this command enter **linkagg slot** followed by the slot number of the NI module and **multiple**.

For example, to enable multiple link aggregation groups on the NI module in Slot 5 enter:

```
-> linkagg slot 5 multiple
```

To configure a first-generation NI module for a single link aggregation group (i.e., the default) use the **linkagg slot single** command. To use this command enter **linkagg slot** followed by the slot number of the NI module and **single**.

For example, to configure a first-generation NI module in Slot 5 for a single link aggregation group enter:

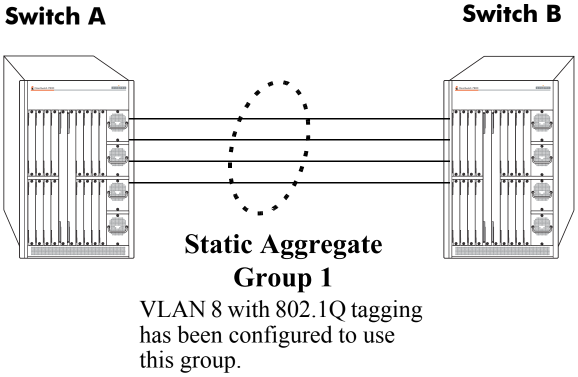
```
-> linkagg slot 5 single
```

When optimization is enabled on a given NI and multiple link aggregation groups are defined, each aggregate group *must* be part of a different VLAN. In other words, traffic cannot be bridged between aggregate groups on the same NI when optimization is enabled. Instead, traffic must be routed.

Application Example

Static link aggregation groups are treated by the switch's software the same way it treats individual physical ports. This section demonstrates this by providing a sample network configuration that uses static link aggregation along with other software features. In addition, a tutorial is provided that shows how to configure this sample network using Command Line Interface (CLI) commands.

The figure below shows VLAN 8, which has been configured on static aggregate 1 and uses 802.1Q tagging. The actual physical links connect ports 4/1, 4/2, 6/1, and 6/2 on Switch A to port 2/1, 2/2, 2/3, and 2/4 on Switch B.


Switch A
Switch B
Static Aggregate Group 1
VLAN 8 with 802.1Q tagging has been configured to use this group.

Sample Network Using Static Link Aggregation

Follow the steps below to configure this network:

Note. Only the steps to configure the local (i.e., Switch A) are provided here since the steps to configure the remote (i.e., Switch B) would not be significantly different.

1 Configure static aggregate group 1 by entering **static linkagg 1 size 4** as shown below:

```
-> static linkagg 1 size 4
```

2 Assign ports 4/1, 4/2, 6/1, and 6/1 to static aggregate group 1 by entering:

```
-> static agg 4/1 agg num 1
-> static agg 4/2 agg num 1
-> static agg 6/1 agg num 1
-> static agg 6/2 agg num 1
```

3 Create VLAN 8 by entering:

```
-> vlan 8
```

4 Configure 802.1Q tagging with a tagging ID of 8 on static aggregate group 1 (on VLAN 8) by entering:

```
-> vlan 8 802.1q 1
```

5 Repeat steps 1 through 4 on Switch B. All the commands would be the same except you would substitute the appropriate port numbers.

Note. *Optional*. Use the **show 802.1q** command to display 802.1Q configurations.

Displaying Static Link Aggregation Configuration and Statistics

You can use Command Line Interface (CLI) **show** commands to display the current configuration and statistics of link aggregation. These commands include the following:

show linkagg Displays information on link aggregation groups.

show linkagg port Displays information on link aggregation ports.

show linkagg slot optimization Displays the link aggregation optimization status and configuration of an NI module.

When you use the **show linkagg** command without specifying the link aggregation group number and when you use the **show linkagg port** command without specifying the slot and port number these commands provide a “global” view of switch-wide link aggregate group and link aggregate port information, respectively.

For example, to display global statistics on all link aggregate groups (both static and dynamic) you would enter

```
-> show linkagg
```

A screen similar to the following would be displayed:

```
Number  Aggregate  SNMP Id  Size  Admin State  Oper State     Att/Sel Ports
-------+----------+--------+----+-------------+-------------+-------------
   1      Static    40000001   8     ENABLED      UP             2   2
   2      Dynamic   40000002   4     ENABLED      DOWN           0   0
   3      Dynamic   40000003   8     ENABLED      DOWN           0   2
   4      Dynamic   40000004   16    ENABLED      UP             3   3
   5      Static    40000005   2     DISABLED     DOWN           0   0
```

When you use the **show linkagg** command with the link aggregation group number and when you use the **show linkagg port** command with the slot and port number these commands provide detailed views of link aggregate group and link aggregate port information, respectively. These detailed views provide excellent tools for diagnosing and troubleshooting problems.

For example, to display detailed statistics for port 1 in slot 4 that is attached to static link aggregate group 1 you would enter:

```
-> show linkagg port 4/1
```

A screen similar to the following would be displayed:

```
Static Aggregable Port
  SNMP Id                        : 4001,
  Slot/Port                      : 4/1,
  Administrative State           : ENABLED,
  Operational State              : DOWN,
  Port State                     : CONFIGURED,
  Link State                     : DOWN,
  Selected Agg Number            : 2,
  Port position in the aggregate : 0,
  Primary port                   : NONE
```

The **show linkagg slot optimization** command provides the status of link aggregation optimization on a single Network Interface (NI) module. For example, to display the optimization status of the NI module in Slot 5 enter:

```
-> show linkagg slot 5 optimization
```

A screen similar to the following will be displayed:

```
Link aggregation optimization is enabled for slot 5
Multiple aggregates per slot disabled for slot 5
```

Note. See the “Link Aggregation Commands” chapter in the *OmniSwitch CLI Reference Guide* for complete documentation of **show** commands for link aggregation.

12 Configuring Dynamic Link Aggregation

Alcatel's dynamic link aggregation software allows you to combine several physical links into one large virtual link known as a link aggregation *group*. Using link aggregation can provide the following benefits:

- **Scalability**. You can configure up to 32 link aggregation groups on an OmniSwitch 7700/7800 or 16 link aggregation groups on an OmniSwitch 8800 that can consist of 2, 4, 8, or 16 10 Mbps, 100 Mbps, or 1 Gbps Ethernet links.
- **Reliability**. If one of the physical links in a link aggregate group goes down (unless it is the last one) the link aggregate group can still operate.
- **Ease of Migration**. Link aggregation can ease the transition from a 100 Mbps Ethernet backbones to Gigabit Ethernet backbones.

Note. This chapter only covers dynamic link aggregation for OmniSwitch 7700, 7800, and 8800 switches. Please refer to the *OmniSwitch 6600 Family Network Configuration Guide* for information on configuring dynamic link aggregation on OmniSwitch 6600 Family switches and the *OmniSwitch 6800 Series Network Configuration Guide* for information on configuring dynamic link aggregation on OmniSwitch 6800 Series switches. These switches which use different procedures and have many different operating ranges.

In This Chapter

This chapter describes the basic components of dynamic link aggregation and how to configure them through the Command Line Interface (CLI). CLI commands are used in the configuration examples; for more details about the syntax of commands, see the *OmniSwitch CLI Reference Guide*.

Configuration procedures described in this chapter include:

- Configuring dynamic link aggregation groups on page 12-10.
- Configuring ports so they can be aggregated in dynamic link aggregation groups on page 12-12.
- Modifying dynamic link aggregation parameters on page 12-14.

Note. You can also configure and monitor dynamic link aggregation with WebView, Alcatel's embedded web-based device management application. WebView is an interactive and easy-to-use GUI that can be launched from OmniVista or a web browser. Please refer to WebView's online documentation for more information on configuring and monitoring dynamic link aggregation with WebView.

Dynamic Link Aggregation Specifications

The table below lists specifications for dynamic aggregation groups and ports:

IEEE Specifications Supported	802.3ad — Aggregation of Multiple Link Segments
Maximum number of link aggregation groups	32 (OmniSwitch 7700/7800) 16 (OmniSwitch 8800)
Number of links per group supported	2, 4, 8, or 16
Range for optional group name	1 to 255 characters
Group actor admin key	0 to 65535
Group actor system priority	0 to 65535
Group partner system priority	0 to 65535
Group partner admin key	0 to 65535
Port actor admin key	0 to 65535
Port actor system priority	0 to 255
Port partner admin key	0 to 65535
Port partner admin system priority	0 to 255
Port actor port	0 to 65535
Port actor port priority	0 to 255
Port partner admin port	0 to 65535
Port partner admin port priority	0 to 255
CLI Command Prefix Recognition	All dynamic link aggregation configuration commands support prefix recognition. (Dynamic link aggregation show commands do not support prefix recognition.) See the "Using the CLI" chapter in the *OmniSwitch 7700/7800/8800 Switch Management Guide* for more information.

Dynamic Link Aggregation Default Values

The table below lists default values for dynamic aggregate groups.

Parameter Description	Command	Default Value/Comments
Group Administrative State	**lacp linkagg admin state**	enabled
Group Name	**lacp linkagg name**	No name configured
Group Actor Administrative Key	**lacp linkagg actor admin key**	0
Group Actor System Priority	**lacp linkagg actor system priority**	0
Group Actor System ID	**lacp linkagg actor system id**	00:00:00:00:00:00
Group Partner System ID	**lacp linkagg partner system id**	00:00:00:00:00:00
Group Partner System Priority	**lacp linkagg partner system priority**	0
Group Partner Administrative Key	**lacp linkagg partner admin key**	0
Actor Port Administrative State	**lacp agg actor admin state**	active timeout aggregate
Actor Port System ID	**lacp agg actor system id**	00:00:00:00:00:00
Partner Port System Administrative State	**lacp agg partner admin state**	active timeout aggregate
Partner Port Admin System ID	**lacp agg partner admin system id**	00:00:00:00:00:00
Partner Port Administrative Key	**lacp agg partner admin key**	0
Partner Port Admin System Priority	**lacp agg partner admin system priority**	0
Actor Port Priority	**lacp agg actor port priority**	0
Partner Port Administrative Port	**lacp agg partner admin port**	0
Partner Port Priority	**lacp agg partner admin port priority**	0
Link aggregation optimization on first-generation NI modules	**linkagg slot optimization**	disabled
Single or multiple link aggregation groups on first-generation NI modules	**linkagg slot single** **linkagg slot multiple**	single

Quick Steps for Configuring Dynamic Link Aggregation

Follow the steps below for a quick tutorial on configuring a dynamic aggregate link between two switches. Additional information on how to configure each command is given in the subsections that follow.

1 Create the dynamic aggregate group on the local (actor) switch with the **lacp linkagg size** command as shown below:

```
-> lacp linkagg 2 size 8 actor admin key 5
```

2 Configure ports (the number of ports should be less than or equal to the size value set in Step 1) with the same actor administrative key (which allows them to be aggregated) with the **lacp agg actor admin key** command. For example:

```
-> lacp agg 1/1 actor admin key 5
-> lacp agg 1/4 actor admin key 5
-> lacp agg 3/3 actor admin key 5
-> lacp agg 5/4 actor admin key 5
-> lacp agg 6/1 actor admin key 5
-> lacp agg 6/2 actor admin key 5
-> lacp agg 7/3 actor admin key 5
-> lacp agg 8/1 actor admin key 5
```

3 Create a VLAN for this dynamic link aggregate group with the **vlan** command. For example:

```
-> vlan 2 port default 2
```

4 Create the equivalent dynamic aggregate group on the remote (partner) switch with the **lacp linkagg size** command as shown below:

```
-> lacp linkagg 2 size 8 actor admin key 5
```

5 Configure ports (the number of ports should be less than or equal to the size value set in Step 4) with the same actor administrative key (which allows them to be aggregated) with the **lacp agg actor admin key** command. For example:

```
-> lacp agg 2/1 actor admin key 5
-> lacp agg 3/1 actor admin key 5
-> lacp agg 3/3 actor admin key 5
-> lacp agg 3/6 actor admin key 5
-> lacp agg 5/1 actor admin key 5
-> lacp agg 5/6 actor admin key 5
-> lacp agg 8/1 actor admin key 5
-> lacp agg 8/3 actor admin key 5
```

6 Create a VLAN for this dynamic link aggregate group with the **vlan** command. For example:

```
-> vlan 2 port default 2
```

Note. As an option, you can verify your dynamic aggregation group settings with the **show linkagg** command on either the actor or partner switch. For example:

```
-> show linkagg 2
Dynamic Aggregate
  SNMP Id                    : 40000002,
  Aggregate Number           : 2,
  SNMP Descriptor            : Dynamic Aggregate Number 2 ref 40000002 size 8,
  Name                       : ,
  Admin State                : ENABLED,
  Operational State          : UP,
  Aggregate Size             : 8,
  Number of Selected Ports : 8,
  Number of Reserved Ports : 8,
  Number of Attached Ports : 8,
  Primary Port               : 1/1,
LACP
  MACAddress                 : [00:1f:cc:00:00:00],
  Actor System Id            : [00:20:da:81:d5:b0],
  Actor System Priority      : 0,
  Actor Admin Key            : 5,
  Actor Oper Key             : 0,
  Partner System Id          : [00:20:da:81:d5:b1],
  Partner System Priority  : 0,
  Partner Admin Key          : 5,
  Partner Oper Key           : 0
```

You can also use the **show linkagg port** port command to display information on specific ports. See "Displaying Dynamic Link Aggregation Configuration and Statistics" on page 12-34 for more information on **show** commands.

An example of what these commands look like entered sequentially on the command line on the actor switch:

```
-> lacp linkagg 2 size 8 actor admin key 5
-> lacp agg 1/1 actor admin key 5
-> lacp agg 1/4 actor admin key 5
-> lacp agg 3/3 actor admin key 5
-> lacp agg 5/4 actor admin key 5
-> lacp agg 6/1 actor admin key 5
-> lacp agg 6/2 actor admin key 5
-> lacp agg 7/3 actor admin key 5
-> lacp agg 8/1 actor admin key 5
-> vlan 2 port default 2
```

An example of what these commands look like entered sequentially on the command line on the partner switch:

```
-> lacp linkagg 2 size 8 actor admin key 5
-> lacp agg 2/1 actor admin key 5
-> lacp agg 3/1 actor admin key 5
-> lacp agg 3/3 actor admin key 5
-> lacp agg 3/6 actor admin key 5
-> lacp agg 5/1 actor admin key 5
-> lacp agg 5/6 actor admin key 5
-> lacp agg 8/1 actor admin key 5
-> lacp agg 8/3 actor admin key 5
-> vlan 2 port default 2
```

Dynamic Link Aggregation Overview

Link aggregation allows you to combine 2, 4, 8, or 16 physical connections into large virtual connections known as link aggregation *groups*. On an OmniSwitch 7700/7800 switch you can create up to 32 link aggregation (both dynamic and static) groups. On an OmniSwitch 8800 switch you can create up to 16 link aggregation (both dynamic and static) groups. You can create Virtual LANs (VLANs), configure Quality of Service (QoS) conditions, 802.1Q framing, and other networking features on link aggregation groups because switch software treats these virtual links just like physical links. (See "Relationship to Other Features" on page 12-9 for more information on how link aggregation interacts with other software features.)

Link aggregation groups are identified by unique MAC addresses, which are created by the switch but can be modified by the user at any time. Load balancing for Layer 2 non IP packets is on a MAC address basis and for IP packets the balancing algorithm uses IP address as well. Ports *must* be the same speed within the same aggregate group.

Alcatel's link aggregation software allows you to configure the following two different types of link aggregation groups:

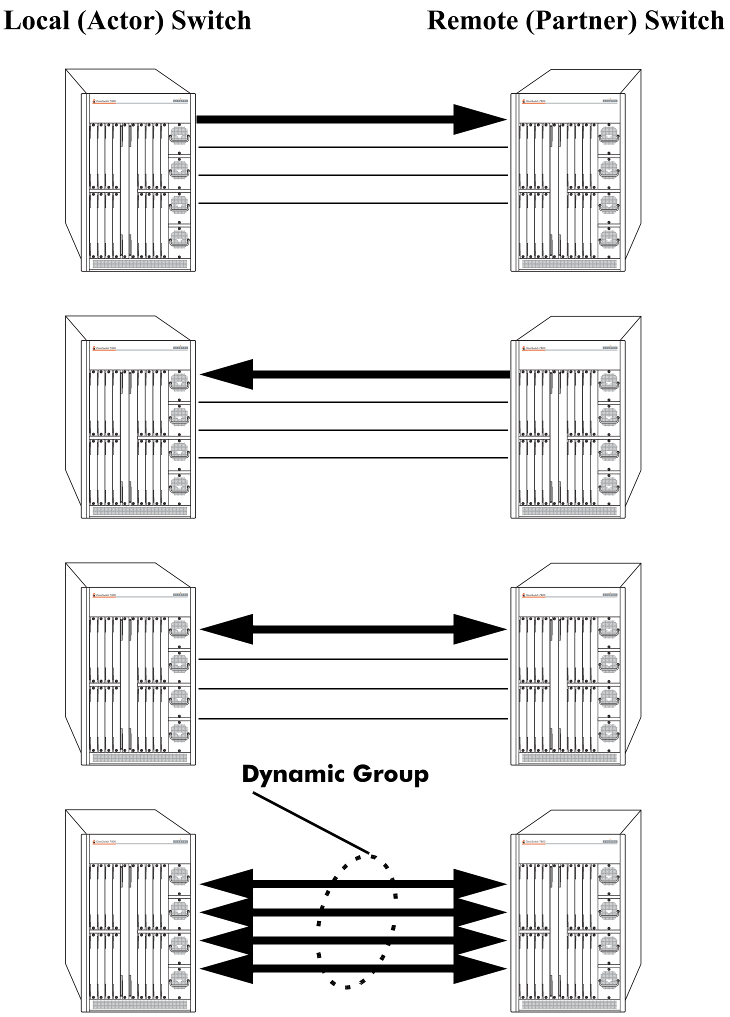
- Static link aggregate groups
- Dynamic link aggregate groups

This chapter describes dynamic link aggregation. For information on static link aggregation, please refer to Chapter 11, "Configuring Static Link Aggregation."

Dynamic Link Aggregation Operation

Dynamic aggregate groups are virtual links between two switches consisting of 2, 4, 8, or 16 10 Mbps, 100 Mbps or 1 Gbps fixed physical links. Dynamic aggregate groups use the standard IEEE 802.3ad Link Aggregation Control Protocol (LACP) to dynamically establish the best possible configuration for the group. This task is accomplished by special Link Aggregation Control Protocol Data Unit (LACPDU) frames that are sent and received by switches on both sides of the link to monitor and maintain the dynamic aggregate group.

The figure on the following page shows a dynamic aggregate group that has been configured between Switch A and Switch B. The dynamic aggregate group links four ports on two OS7-GNI-U2 modules on Switch A to four ports on two OS7-GNI-U2 modules on Switch B.



1. Local (actor) switch sends requests to establish a dynamic aggregate group link to the remote (partner) switch.

2. Partner switch acknowledges that it can accept this dynamic group.

3. Actor and partner switches negotiate parameters for the dynamic group, producing optimal settings.

4. Actor and partner switches establish the dynamic aggregate group. LACPDU messages are sent back and forth to monitor and maintain the group.

Example of a Dynamic Aggregate Group Network

Dynamic aggregate groups can be created between two OmniSwitch 7700/7800/8800 switches or between an OmniSwitch 7700/7800/8800 and another vendor's switch if that vendor supports IEEE 802.3ad LACP.

See "Configuring Dynamic Link Aggregate Groups" on page 12-10 for information on using Command Line Interface (CLI) commands to configure dynamic aggregate groups and see "Displaying Dynamic Link Aggregation Configuration and Statistics" on page 12-34 for information on using the CLI to monitor dynamic aggregate groups.

Relationship to Other Features

Link aggregation groups are supported by other switch software features. For example, you can configure 802.1Q tagging on link aggregation groups in addition to configuring it on individual ports. The following features have CLI commands or command parameters that support link aggregation:

- **VLANs**. For more information on VLANs see Chapter 5, “Configuring VLANs.”
- **802.1Q**. For more information on configuring and monitoring 802.1Q see Chapter 10, “Configuring 802.1Q.”
- **Spanning Tree**. For more information on Spanning Tree see Chapter 6, “Configuring Spanning Tree Parameters.”

Note. See “Application Examples” on page 12-30 for tutorials on using link aggregation with other features.

Configuring Dynamic Link Aggregate Groups

This section describes how to use Alcatel's Command Line Interface (CLI) commands to create, modify, and delete dynamic aggregate groups. See "Configuring Mandatory Dynamic Link Aggregate Parameters" on page 12-10 for more information.

Note. See "Quick Steps for Configuring Dynamic Link Aggregation" on page 12-4 for a brief tutorial on configuring these mandatory parameters.

Alcatel's link aggregation software is preconfigured with the default values for dynamic aggregate groups and ports shown in the table in "Dynamic Link Aggregation Default Values" on page 12-3. For most configurations, using only the steps described in "Creating and Deleting a Dynamic Aggregate Group" on page 12-11 will be necessary to configure a dynamic link aggregate group. However, if you need to modify any of the parameters listed in the table on page 12-3, please see "Modifying Dynamic Link Aggregate Group Parameters" on page 12-14 for more information.

Note. See the "Link Aggregation Commands" chapter in the *OmniSwitch CLI Reference Guide* for complete documentation of **show** commands for link aggregation.

Configuring Mandatory Dynamic Link Aggregate Parameters

When configuring dynamic link aggregates on a switch you must perform the following steps:

Note. Depending on the hardware version of your Network Interface (NI) modules, you may need to optimize those modules for link aggregation. See "Optimizing NI Modules for Link Aggregation" on page 12-29 for more information. In addition, if you optimize an NI for link aggregation only one link aggregation group can be configured on this module. To configure additional link aggregation groups see "Configuring Multiple Link Aggregation Groups on NI Modules" on page 12-29 for more information.

1 Create the Dynamic Aggregate Groups on the Local (Actor) and Remote (Partner) Switches. To create a dynamic aggregate group use the **lacp linkagg size** command, which is described in "Creating and Deleting a Dynamic Aggregate Group" on page 12-11.

2 Configure the Same Administrative Key on the Ports You Want to Join the Dynamic Aggregate Group. To configure ports with the same administrative key (which allows them to be aggregated) use the **lacp agg actor admin key** command, which is described in "Assigning and Removing Ports in a Dynamic Aggregate Group" on page 12-12.

Note. Depending on the needs of your network you may need to configure additional parameters. Commands to configure optional dynamic link aggregate parameters are described in "Modifying Dynamic Link Aggregate Group Parameters" on page 12-14. These commands must be executed after you create a dynamic aggregate group.

Creating and Deleting a Dynamic Aggregate Group

The following subsections describe how to create and delete dynamic aggregate groups with the **lacp linkagg size** command.

Creating a Dynamic Aggregate Group

To configure a dynamic aggregate group enter **lacp linkagg**, followed by the user-configured dynamic aggregate number (which can be 0 through 31 on an OmniSwitch 7700/7800 switch and 0 through 15 on an OmniSwitch 8800 switch), **size**, and the maximum number of links that will belong to this dynamic aggregate group. For example, to configure dynamic aggregate group 2 consisting of eight (8) links enter:

```
-> lacp linkagg 2 size 8
```

You can create up to 32 aggregation groups (both static and dynamic) on an OmniSwitch 7700/7800 switch and up to 16 aggregation groups (both static and dynamic) on an OmniSwitch 8800 switch. However, due to the number of hardware queues available for link aggregation, the maximum number of link aggregate groups decreases as the aggregate group size (i.e., number of links) increases. See the table below for more information on OmniSwitch 7700/7800.

Number of Links (Aggregate Size) on OmniSwitch 7700/7800	Maximum Number of Aggregate Groups on a Switch
2	32 (range 0–31)
4	29 (range 0–28)
8	14 (range 0–13)
16	7 (range 0–6)

And see the table below for OmniSwitch 8800:

Number of Links (Aggregate Size) on OmniSwitch 8800	Maximum Number of Aggregate Groups on a Switch
2	16 (range 0–15)
4	16 (range 0–15)
8	14 (range 0–13)
16	7 (range 0–6)

You can also specify optional parameters shown in the table below. These parameters must be entered after **size** and the user-specified number of links.

lacp linkagg size keywords		
name	**admin state enable**	**partner admin key**
actor system priority	**admin state disable**	**actor admin key**
partner system priority	**actor system id**	**partner system id**

For example, Alcatel recommends assigning the actor admin key when you create the dynamic aggregate group to help ensure that ports are assigned to the correct group. To create a dynamic aggregate group with aggregate number 3 consisting of two ports with an admin actor key of 10, for example, enter:

```
-> lacp linkagg 3 size 2 actor admin key 10
```

Note. The optional keywords for this command may be entered in any order as long as they are entered after **size** and the user-specified number of links.

Deleting a Dynamic Aggregate Group

To remove a dynamic aggregation group configuration from a switch use the **no** form of the **lacp linkagg size** command by entering **no lacp linkagg** followed by its dynamic aggregate group number.

For example, to delete dynamic aggregate group 2 from a switch's configuration you would enter:

```
-> no lacp linkagg 2
```

Note. You cannot delete a dynamic aggregate group if it has any attached ports. To remove attached ports you must disable the dynamic aggregate group with the **lacp linkagg admin state** command, which is described in "Disabling a Dynamic Aggregate Group" on page 12-15.

Assigning and Removing Ports in a Dynamic Aggregate Group

The following subsections describe how to configure ports with the same administrative key (which allows them to be aggregated) or to remove them from a dynamic aggregate group with the **lacp agg actor admin key** command.

Note. Link Aggregation is not supported on OS8-10GNI-UR1 modules.

Configuring Ports To Join a Dynamic Aggregate Group

To configure ports with the same administrative key (which allows them to be aggregated) enter **lacp agg** followed by the slot number, a slash (/), the port number, **actor admin key**, and the user-specified actor administrative key (which can range from 0 to 65535).

Note. A port may belong to only one aggregate group. In addition, mobile ports cannot be aggregated. See Chapter 7, "Assigning Ports to VLANs," for more information on mobile ports.

For example, to configure port 1 in slot 4 with an administrative key of 10 you would enter:

```
-> lacp agg 4/1 actor admin key 10
```

A port may belong to only one aggregate group. In addition, mobile ports cannot be aggregated.

Note. When assigning ports the same actor administrative key to multiple ports keep in mind that the ports do not need to be sequentially located and can also be on different Network Interfaces (NIs).

You must execute the **lacp agg actor admin key** command on all ports in a dynamic aggregate group. If not, the ports will be unable to join the group.

You can also specify optional parameters shown in the table below. These keywords must be entered after actor admin key and the user-specified actor administrative key value.

lacp agg actor admin key keywords		
actor admin state **actor system priority** **partner admin system priority** **partner admin port priority**	**partner admin state** **partner admin system id** **actor port priority**	**actor system id** **partner admin key** **partner admin port**

Note. The **actor admin state** and **partner admin state** keywords have additional parameters, which are described in "Modifying the Actor Port System Administrative State" on page 12-19 and "Modifying the Partner Port System Administrative State" on page 12-23, respectively.

All of the optional keywords listed above for this command may be entered in any order as long as they appear after the **actor admin key** keywords and their user-specified value.

For example, to assign actor administrative key of 10, a local system ID (MAC address) of 00:20:da:06:ba:d3, and a local priority of 200 to slot 4 port 1, enter:

```
-> lacp agg 4/1 actor admin key 10 actor system id 00:20:da:06:ba:d3 actor
system priority 200
```

As an option, you can use the **ethernet**, **fastethernet**, and **gigaethernet** keywords before the slot and port number to document the interface type to make the command look consistent with early-generation Alcatel CLI syntax. For example, to assign actor administrative key of 10 and to document that the port is a 10 Mbps Ethernet port to slot 4 port 1, enter:

```
-> lacp agg ethernet 4/1 actor admin key 10
```

Note. The **ethernet**, **fastethernet**, and **gigaethernet** keywords do not modify a port's configuration. See Chapter 1, "Configuring Ethernet Ports," for information on configuring Ethernet ports.

Removing Ports from a Dynamic Aggregate Group

To remove a port from a dynamic aggregate group, use the **no** form of the **lacp agg actor admin key** command by entering **lacp agg no** followed by the slot number, a slash (/), and the port number.

For example, to remove port 1 in slot 4 from any dynamic aggregate group you would enter:

```
-> lacp agg no 4/1
```

Modifying Dynamic Link Aggregate Group Parameters

The table on page 12-3 lists default group and port settings for Alcatel's dynamic link aggregation software. These parameters ensure compliance with the IEEE 802.3ad specification. For most networks, these default values do not need to be modified or will be modified automatically by switch software. However, if you need to modify any of these default settings see the following sections to modify parameters for:

- Dynamic aggregate groups beginning on page 12-14
- Dynamic aggregate actor ports beginning on page 12-18
- Dynamic aggregate partner ports beginning on page 12-23.

Note. You *must* create a dynamic aggregate group before you can modify group or port parameters. See "Configuring Dynamic Link Aggregate Groups" on page 12-10 for more information.

Modifying Dynamic Aggregate Group Parameters

This section describes how to modify the following dynamic aggregate group parameters:

- Group name (see "Modifying the Dynamic Aggregate Group Name" on page 12-14)
- Group administrative state (see "Modifying the Dynamic Aggregate Group Administrative State" on page 12-15)
- Group local (actor) switch actor administrative key (see "Configuring and Deleting the Dynamic Aggregate Group Actor Administrative Key" on page 12-15)
- Group local (actor) switch system priority (see "Modifying the Dynamic Aggregate Group Actor System Priority" on page 12-16)
- Group local (actor) switch system ID (see "Modifying the Dynamic Aggregate Group Actor System ID" on page 12-16)
- Group remote (partner) administrative key (see "Modifying the Dynamic Aggregate Group Partner Administrative Key" on page 12-17)
- Group remote (partner) system priority (see "Modifying the Dynamic Aggregate Group Partner System Priority" on page 12-17)
- Group remote (partner) switch system ID (see "Modifying the Dynamic Aggregate Group Partner System ID" on page 12-18)

Modifying the Dynamic Aggregate Group Name

The following subsections describe how to configure and remove a dynamic aggregate group name with the **lacp linkagg name** command.

Configuring a Dynamic Aggregate Group name

To configure a dynamic aggregate group name enter **lacp linkagg** followed by the dynamic aggregate group number, **name**, and the user-specified name, which can be from 1 to 255 characters long.

For example, to name dynamic aggregate group 4 “Engineering” you would enter:

```
-> lacp linkagg 4 name Engineering
```

Note. If you want to specify spaces within a name, the name must be enclosed with quotes. For example:

```
-> lacp linkagg 4 name "Engineering Lab"
```

Deleting a Dynamic Aggregate Group Name

To remove a dynamic aggregate group name from a switch’s configuration use the **no** form of the **lacp linkagg name** command by entering **lacp linkagg** followed by the dynamic aggregate group number and **no name**.

For example, to remove any user-configured name from dynamic aggregate group 4 you would enter:

```
-> lacp linkagg 4 no name
```

Modifying the Dynamic Aggregate Group Administrative State

By default, the dynamic aggregate group administrative state is enabled. The following subsections describe how to enable and disable a dynamic aggregate group’s administrative state with the **lacp linkagg admin state** command.

Enabling a Dynamic Aggregate Group

To enable the dynamic aggregate group administrative state enter **lacp linkagg** followed by the dynamic aggregate group number and **admin state enable**. For example, to enable dynamic aggregate group 4 you would enter:

```
-> lacp linkagg 4 admin state enable
```

Disabling a Dynamic Aggregate Group

To disable a dynamic aggregate group’s administrative state use the **lacp linkagg admin state** command by entering **lacp linkagg** followed by the dynamic aggregate group number and **admin state disable**.

For example, to disable dynamic aggregate group 4 you would enter:

```
-> lacp linkagg 4 admin state disable
```

Configuring and Deleting the Dynamic Aggregate Group Actor Administrative Key

The following subsections describe how to configure and delete a dynamic aggregate group actor administrative key with the **lacp linkagg actor admin key** command.

Configuring a Dynamic Aggregate Actor Administrative Key

To configure the dynamic aggregate group actor switch administrative key enter **lacp linkagg** followed by the dynamic aggregate group number, **actor admin key**, and the value for the administrative key, which can be 0 through 65535.

For example, to configure dynamic aggregate group 4 with an administrative key of 10 you would enter:

```
-> lacp linkagg 4 actor admin key 10
```

Deleting a Dynamic Aggregate Actor Administrative Key

To remove an actor switch administrative key from a dynamic aggregate group's configuration use the **no** form of the **lacp linkagg actor admin key** command by entering **lacp linkagg** followed by the dynamic aggregate group number, and **no actor admin key**.

For example, to remove an administrative key from dynamic aggregate group 4 you would enter:

```
-> lacp linkagg 4 no actor admin key
```

Modifying the Dynamic Aggregate Group Actor System Priority

By default, the dynamic aggregate group actor system priority is 0. The following subsections describe how to configure a user-specified value and how to restore the value to its default value with the **lacp linkagg actor system priority** command.

Configuring a Dynamic Aggregate Group Actor System Priority

You can configure a user-specified dynamic aggregate group actor system priority value to a value ranging from 0 to 65535 by entering **lacp linkagg** followed by the dynamic aggregate group number, **actor system priority**, and the new priority value.

For example, to change the actor system priority of dynamic aggregate group 4 to 2000 you would enter:

```
-> lacp linkagg 4 actor system priority 2000
```

Restoring the Dynamic Aggregate Group Actor System Priority

To restore the dynamic aggregate group actor system priority to its default (i.e., 0) value use the **no** form of the **lacp linkagg actor system priority** command by entering **lacp linkagg** followed by the dynamic aggregate group number and **no actor system priority**.

For example, to restore the actor system priority to its default value on dynamic aggregate group 4 you would enter:

```
-> lacp linkagg 4 no actor system priority
```

Modifying the Dynamic Aggregate Group Actor System ID

By default, the dynamic aggregate group actor system ID (MAC address) is 00:00:00:00:00:00. The following subsections describe how to configure a user-specified value and how to restore the value to its default value with the **lacp linkagg actor system id** command.

Configuring a Dynamic Aggregate Group Actor System ID

You can configure a user-specified dynamic aggregate group actor system ID by entering **lacp linkagg** followed by the dynamic aggregate group number, **actor system id**, and the user-specified MAC address (in the hexadecimal format of *xx:xx:xx:xx:xx:xx*), which is used as the system ID.

For example, to configure the system ID on dynamic aggregate group 4 as 00:20:da:81:d5:b0 you would enter:

```
-> lacp linkagg 4 actor system id 00:20:da:81:d5:b0
```

Restoring the Dynamic Aggregate Group Actor System ID

To remove the user-configured actor switch system ID from a dynamic aggregate group's configuration use the **no** form of the **lacp linkagg actor system id** command by entering **lacp linkagg** followed by the dynamic aggregate group number and **no actor system id**.

For example, to remove the user-configured system ID from dynamic aggregate group 4 you would enter:

```
-> lacp linkagg 4 no actor system id
```

Modifying the Dynamic Aggregate Group Partner Administrative Key

By default, the dynamic aggregate group partner administrative key (i.e., the administrative key of the partner switch) is 0. The following subsections describe how to configure a user-specified value and how to restore the value to its default value with the **lacp linkagg partner admin key** command.

Configuring a Dynamic Aggregate Group Partner Administrative Key

You can modify the dynamic aggregate group partner administrative key to a value ranging from 0 to 65535 by entering **lacp linkagg** followed by the dynamic aggregate group number, **partner admin key**, and the value for the administrative key, which can be 0 through 65535.

For example, to set the partner administrative key to 4 on dynamic aggregate group 4 you would enter:

```
-> lacp linkagg 4 partner admin key 10
```

Restoring the Dynamic Aggregate Group partner Administrative Key

To remove a partner administrative key from a dynamic aggregate group's configuration use the **no** form of the **lacp linkagg partner admin key** command by entering **lacp linkagg** followed by the dynamic aggregate group number, and **no partner admin key**.

For example, to remove the user-configured partner administrative key from dynamic aggregate group 4 you would enter:

```
-> lacp linkagg 4 no partner admin key
```

Modifying the Dynamic Aggregate Group Partner System Priority

By default, the dynamic aggregate group partner system priority is 0. The following subsections describe how to configure a user-specified value and how to restore the value to its default value with the **lacp linkagg partner system priority** command.

Configuring a Dynamic Aggregate Group Partner System Priority

You can modify the dynamic aggregate group partner system priority to a value ranging from 0 to 65535 by entering **lacp linkagg** followed by the dynamic aggregate group number, **partner system priority**, and the new priority value.

For example, to set the partner system priority on dynamic aggregate group 4 to 2000 you would enter:

```
-> lacp linkagg 4 partner system priority 2000
```

Restoring the Dynamic Aggregate Group Partner System Priority

To restore the dynamic aggregate group partner system priority to its default (i.e., 0) value use the **no** form of the **lacp linkagg partner system priority** command by entering **lacp linkagg** followed by the dynamic aggregate group number and **no partner system priority**.

For example, to reset the partner system priority of dynamic aggregate group 4 to its default value you would enter:

```
-> lacp linkagg 4 no partner system priority
```

Modifying the Dynamic Aggregate Group Partner System ID

By default, the dynamic aggregate group partner system ID is 00:00:00:00:00:00. The following subsections describe how to configure a user-specified value and how to restore the value to its default value with the **lacp linkagg partner system id** command.

Configuring a Dynamic Aggregate Group Partner System ID

You can configure the dynamic aggregate group partner system ID by entering **lacp linkagg** followed by the dynamic aggregate group number, **partner system id**, and the user-specified MAC address (in the hexadecimal format of *xx:xx:xx:xx:xx:xx*), which is used as the system ID.

For example, to configure the partner system ID as 00:20:da:81:d5:b0 on dynamic aggregate group 4 you would enter:

```
-> lacp linkagg 4 partner system id 00:20:da:81:d5:b0
```

Restoring the Dynamic Aggregate Group Partner System ID

To remove the user-configured partner switch system ID from the dynamic aggregate group's configuration use the **no** form of the **lacp linkagg partner system id** command by entering **lacp linkagg** followed by the dynamic aggregate group number and **no partner system id**.

For example, to remove the user-configured partner system ID from dynamic aggregate group 4 you would enter:

```
-> lacp linkagg 4 no partner system id
```

Modifying Dynamic Link Aggregate Actor Port Parameters

This section describes how to modify the following dynamic aggregate actor port parameters:

- Actor port administrative state (see "Modifying the Actor Port System Administrative State" on page 12-19)
- Actor port system ID (see "Modifying the Actor Port System ID" on page 12-20)
- Actor port system priority (see "Modifying the Actor Port System Priority" on page 12-21)
- Actor port priority (see "Modifying the Actor Port Priority" on page 12-22)

Note. See "Assigning and Removing Ports in a Dynamic Aggregate Group" on page 12-12 for information on modifying a dynamic aggregate group administrative key.

All of the commands to modify actor port parameters allow you to add the **ethernet**, **fastethernet**, and **gigaethernet** keywords before the slot and port number to document the interface type or make the command look consistent with early-generation Alcatel CLI syntax. However, these keywords do not modify a port's configuration. See Chapter 1, "Configuring Ethernet Ports," for information on configuring Ethernet ports.

Note. A port may belong to only one aggregate group. In addition, mobile ports cannot be aggregated. See Chapter 7, "Assigning Ports to VLANs," for more information on mobile ports.

Modifying the Actor Port System Administrative State

The system administrative state of a dynamic aggregate group actor port is indicated by bit settings in Link Aggregation Control Protocol Data Unit (LACPDU) frames sent by the port. By default, bits 0 (indicating that the port is active), 1 (indicating that short timeouts are used for LACPDU frames), and 2 (indicating that this port is available for aggregation) are set in LACPDU frames.

The following subsections describe how to configure user-specified values and how to restore them to their default values with the **lacp agg actor admin state** command.

Configuring Actor Port Administrative State Values

To configure an LACP actor port's system administrative state values by entering **lacp agg**, the slot number, a slash (/), the port number, **actor admin state,** and one or more of the keywords shown in the table below *or* **none**:

lacp agg actor admin state Keyword	Definition
active	Specifies that bit 0 in LACPDU frames is set, which indicates that the link is able to exchange LACPDU frames. By default, this bit is set.
timeout	Specifies that bit 1 in LACPDU frames is set, which indicates that a short timeout is used for LACPDU frames. When this bit is disabled, a long timeout is used for LACPDU frames. By default, this bit is set.
aggregate	Specifies that bit 2 in LACPDU frames is set, which indicates that the system considers this link to be a potential candidate for aggregation. If this bit is not set, the system considers the link to be individual (it can only operate as a single link). By default, this bit is set.
synchronize	Specifying this keyword has no effect because the system always determines its value. When this bit (bit 3) is set by the system, the port is allocated to the correct dynamic aggregation group. If this bit is not set by the system, the port is not allocated to the correct dynamic aggregation group.
collect	Specifying this keyword has no effect because the system always determines its value. When this bit (bit 4) is set by the system, incoming LACPDU frames are collected from the individual ports that make up the dynamic aggregate group.
distribute	Specifying this keyword has no effect because the system always determines its value. When this bit (bit 5) is set by the system, distributing outgoing frames on the port is disabled.
default	Specifying this keyword has no effect because the system always determines its value. When this bit (bit 6) is set by the system, it indicates that the actor is using defaulted partner information administratively configured for the partner.

lacp agg actor admin state Keyword	Definition
expire	Specifying this keyword has no effect because the system always determines its value. When this bit (bit 7) is set by the system, the actor cannot receive LACPDU frames.

Note. Specifying **none** removes all administrative states from the LACPDU configuration. For example:

```
-> lacp agg 5/49 actor admin state none
```

For example, to set bits 0 (**active**) and 2 (**aggregate**) on dynamic aggregate actor port 49 in slot 5 you would enter:

```
-> lacp agg 5/49 actor admin state active aggregate
```

As an option you can use the **ethernet**, **fastethernet**, and **gigaethernet** keywords before the slot and port number to document the interface type or make the command look consistent with early-generation Alcatel CLI syntax. For example, to set bits 0 (**active**) and 2 (**aggregate**) on dynamic aggregate actor port 49 in slot 5 and document that the port is a Gigabit Ethernet port you would enter:

```
-> lacp agg gigaethernet 5/49 actor admin state active aggregate
```

Restoring Actor Port Administrative State Values

To restore LACPDU bit settings to their default values use **lacp agg actor admin state** command by entering enter **no** before the **active**, **timeout**, and **aggregate** keywords.

For example, to restore bits 0 (**active**) and 2 (**aggregate**) to their default settings on dynamic aggregate actor port 2 in slot 5 you would enter:

```
-> lacp agg 5/2 actor admin state no active no aggregate
```

Note. Since individual bits with the LACPDU frame are set with the **lacp agg actor admin state** command you can set some bits on and restore other bits within the same command. For example, if you wanted to restore bit 2 (**aggregate**) to its default settings and set bit 0 (**active**) on dynamic aggregate actor port 49 in slot 5 you would enter:

```
-> lacp agg 5/49 actor admin state active no aggregate
```

Modifying the Actor Port System ID

By default, the actor port system ID (i.e., the MAC address used as the system ID on dynamic aggregate actor ports) is 00:00:00:00:00:00. The following subsections describe how to configure a user-specified value and how to restore the value to its default value with the **lacp agg actor system id** command.

Configuring an Actor Port System ID

You can configure the actor port system ID by entering **lacp agg**, the slot number, a slash (/), the port number, **actor system id**, and the user specified actor port system ID (i.e., MAC address) in the hexadecimal format of xx:xx:xx:xx:xx:xx.

For example, to modify the system ID of dynamic aggregate actor port 3 in slot 7 to **00:20:da:06:ba:d3** you would enter:

```
-> lacp agg 7/3 actor system id 00:20:da:06:ba:d3
```

As an option, you can use the **ethernet**, **fastethernet**, and **gigaethernet** keywords before the slot and port number to document the interface type or make the command look consistent with early-generation Alcatel CLI syntax. For example, to modify the system ID of dynamic aggregate actor port 3 in slot 7 to **00:20:da:06:ba:d3** and document that the port is 10 Mbps Ethernet you would enter:

```
-> lacp agg ethernet 7/3 actor system id 00:20:da:06:ba:d3
```

Restoring the Actor Port System ID

To remove a user-configured system ID from a dynamic aggregate group actor port's configuration use the **no** form of the **lacp agg actor system id** command by entering **lacp agg**, the slot number, a slash (/), the port number, and **no actor system id**.

For example, to remove a user-configured system ID from dynamic aggregate actor port 3 in slot 7 you would enter:

```
-> lacp agg 7/3 no actor system id
```

Modifying the Actor Port System Priority

By default, the actor system priority is 0. The following subsections describe how to configure a user-specified value and how to restore the value to its default value with the **lacp agg actor system priority** command.

Configuring an Actor Port System Priority

You can configure the actor system priority to a value ranging from 0 to 255 by entering **lacp agg**, the slot number, a slash (/), the port number, **actor system priority**, and the user-specified actor port system priority.

For example, to modify the system priority of dynamic aggregate actor port 5 in slot 2 to 200 you would enter:

```
-> lacp agg 2/5 actor system priority 200
```

As an option, you can use the **ethernet**, **fastethernet**, and **gigaethernet** keywords before the slot and port number to document the interface type or make the command look consistent with early-generation Alcatel CLI syntax. For example, to modify the system priority of dynamic aggregate actor port 5 in slot 2 to 200 and document that the port is a fast Ethernet port you would enter:

```
-> lacp agg fastethernet 2/5 actor system priority 200
```

Restoring the Actor Port System Priority

To remove a user-configured actor port system priority from a dynamic aggregate group actor port's configuration use the **no** form of the **lacp agg actor system priority** command by entering **lacp agg**, the slot number, a slash (/), the port number, and **no actor system priority**.

For example, to remove a user-configured system priority from dynamic aggregate actor port 5 in slot 2 you would enter:

```
-> lacp agg 2/5 no actor system priority
```

Modifying the Actor Port Priority

By default, the actor port priority (used to converge dynamic key changes) is 0. The following subsections describe how to configure a user-specified value and how to restore the value to its default value with the **lacp agg actor port priority** command.

Configuring the Actor Port Priority

You can configure the actor port priority to a value ranging from 0 to 255 by entering **lacp agg**, the slot number, a slash (/), the port number, **actor port priority**, and the user-specified actor port priority.

For example, to modify the actor port priority of dynamic aggregate actor port 1 in slot 2 to 100 you would enter:

```
-> lacp agg 2/1 actor port priority 100
```

As an option, you can use the **ethernet**, **fastethernet**, and **gigaethernet** keywords before the slot and port number to document the interface type or make the command look consistent with early-generation Alcatel CLI syntax. For example, to modify the actor port priority of dynamic aggregate actor port 1 in slot 2 to 100 and document that the port is a fast Ethernet port you would enter:

```
-> lacp agg fastethernet 2/1 actor port priority 100
```

Restoring the Actor Port Priority

To remove a user configured actor port priority from a dynamic aggregate group actor port's configuration use the **no** form of the **lacp agg actor port priority** command by entering **lacp agg**, the slot number, a slash (/), the port number, and **no actor port priority**.

For example, to remove a user-configured actor priority from dynamic aggregate actor port 1 in slot 2 you would enter:

```
-> lacp agg 2/1 no actor port priority
```

Modifying Dynamic Aggregate Partner Port Parameters

This section describes how to modify the following dynamic aggregate partner port parameters:

- Partner port system administrative state (see "Modifying the Partner Port System Administrative State" on page 12-23)
- Partner port administrative key (see "Modifying the Partner Port Administrative Key" on page 12-25)
- Partner port system ID (see "Modifying the Partner Port System ID" on page 12-26)
- Partner port system priority (see "Modifying the Partner Port System Priority" on page 12-26)
- Partner port administrative state (see "Modifying the Partner Port Administrative Status" on page 12-27)
- Partner port priority (see "Modifying the Partner Port Priority" on page 12-28)

All of the commands to modify partner port parameters allow you to add the **ethernet**, **fastethernet**, and **gigaethernet** keywords before the slot and port number to document the interface type or make the command look consistent with early-generation Alcatel CLI syntax. However, these keywords do not modify a port's configuration. See Chapter 1, "Configuring Ethernet Ports," for information on configuring Ethernet ports.

Note. A port may belong to only one aggregate group. In addition, mobile ports cannot be aggregated. See Chapter 7, "Assigning Ports to VLANs," for more information on mobile ports.

Modifying the Partner Port System Administrative State

The system administrative state of a dynamic aggregate group partner (i.e., remote switch) port is indicated by bit settings in Link Aggregation Control Protocol Data Unit (LACPDU) frames sent by this port. By default, bits 0 (indicating that the port is active), 1 (indicating that short timeouts are used for LACPDU frames), and 2 (indicating that this port is available for aggregation) are set in LACPDU frames.

The following subsections describe how to configure user-specified values and how to restore them to their default values with the **lacp agg partner admin state** command.

Configuring Partner Port System Administrative State Values

To configure the dynamic aggregate partner port's system administrative state values by entering **lacp agg**, the slot number, a slash (/), the port number, **partner admin state,** and one or more of the keywords shown in the table below *or* **none**:

Keyword	Definition
active	Specifies that bit 0 in LACPDU frames is set, which indicates that the link is able to exchange LACPDU frames. By default, this bit is set.
timeout	Specifies that bit 1 in LACPDU frames is set, which indicates that a short timeout is used for LACPDU frames. When this bit is disabled, a long timeout is used for LACPDU frames. By default, this bit is set.
aggregate	Specifies that bit 2 in LACPDU frames is set, which indicates that the system considers this link to be a potential candidate for aggregation. If this bit is not set, the system considers the link to be individual (it can only operate as a single link). By default, this bit is set.

Keyword	Definition
synchronize	Specifies that bit 3 in the partner state octet is enabled. When this bit is set, the port is allocated to the correct dynamic aggregation group. If this bit is not enabled, the port is not allocated to the correct aggregation group. By default, this value is disabled.
collect	Specifying this keyword has no effect because the system always determines its value. When this bit (bit 4) is set by the system, incoming LACPDU frames are collected from the individual ports that make up the dynamic aggregate group.
distribute	Specifying this keyword has no effect because the system always determines its value. When this bit (bit 5) is set by the system, distributing outgoing frames on the port is disabled.
default	Specifying this keyword has no effect because the system always determines its value. When this bit (bit 6) is set by the system, it indicates that the partner is using defaulted actor information administratively configured for the partner.
expire	Specifying this keyword has no effect because the system always determines its value. When this bit (bit 7) is set by the system, the actor cannot receive LACPDU frames.

Note. Specifying **none** removes all administrative states from the LACPDU configuration. For example:

```
-> lacp agg 7/49 partner admin state none
```

For example, to set bits 0 (**active**) and 2 (**aggregate**) on dynamic aggregate partner port 49 in slot 7 you would enter:

```
-> lacp agg 7/49 partner admin state active aggregate
```

As an option, you can use the **ethernet**, **fastethernet**, and **gigaethernet** keywords before the slot and port number to document the interface type or make the command look consistent with early-generation Alcatel CLI syntax. For example, to set bits 0 (**active**) and 2 (**aggregate**) on dynamic aggregate partner port 49 in slot 7 and document that the port is a Gigabit Ethernet port you would enter:

```
-> lacp agg gigaethernet 7/49 partner admin state active aggregate
```

Restoring Partner Port System Administrative State Values

To restore LACPDU bit settings to their default values use the **no** form of the **lacp agg partner admin state** command by entering **no** before the **active**, **timeout**, **aggregate**, or **synchronize** keywords.

For example, to restore bits 0 (**active**) and 2 (**aggregate**) to their default settings on dynamic aggregate partner port 1 in slot 7 you would enter:

```
-> lacp agg 7/1 partner admin state no active no aggregate
```

Note. Since individual bits with the LACPDU frame are set with the **lacp agg partner admin state** command you can set some bits on and restore other bits to default values within the same command. For example, if you wanted to restore bit 2 (**aggregate**) to its default settings and set bit 0 (**active**) on dynamic aggregate partner port 1 in slot 7 you would enter:

```
-> lacp agg 7/1 partner admin state active no aggregate
```

Modifying the Partner Port Administrative Key

By default, the dynamic aggregate partner port's administrative key is 0. The following subsections describe how to configure a user-specified value and how to restore the value to its default value with the **lacp agg partner admin key** command.

Configuring the Partner Port Administrative Key

You can configure the dynamic aggregate partner port's administrative key to a value ranging from 0 to 65535 by entering **lacp agg**, the slot number, a slash (/), the port number, **partner admin key**, and the user-specified partner port administrative key.

For example, to modify the administrative key of a dynamic aggregate group partner port 1 in slot 6 to 1000 enter:

```
-> lacp agg 6/1 partner admin key 1000
```

As an option, you can use the **ethernet**, **fastethernet**, and **gigaethernet** keywords before the slot and port number to document the interface type or make the command look consistent with early-generation Alcatel CLI syntax. For example, to modify the administrative key of a dynamic aggregate group partner port 1 in slot 6 to 1000 and document that the port is a 10 Mbps Ethernet port you would enter:

```
-> lacp agg ethernet 6/1 partner admin key 1000
```

Restoring the Partner Port Administrative Key

To remove a user-configured administrative key from a dynamic aggregate group partner port's configuration use the **no** form of the **lacp agg partner admin key** command by entering **lacp agg**, the slot number, a slash (/), the port number, and **no partner admin key**.

For example, to remove the user-configured administrative key from dynamic aggregate partner port 1 in slot 6, enter:

```
-> lacp agg 6/1 no partner admin key
```

Modifying the Partner Port System ID

By default, the partner port system ID (i.e., the MAC address used as the system ID on dynamic aggregate partner ports) is 00:00:00:00:00:00. The following subsections describe how to configure a user-specified value and how to restore the value to its default value with the **lacp agg partner admin system id** command.

Configuring the Partner Port System ID

You can configure the partner port system ID by entering **lacp agg**, the slot number, a slash (/), the port number, **partner admin system id**, and the user-specified partner administrative system ID (i.e., the MAC address in hexadecimal format).

For example, to modify the system ID of dynamic aggregate partner port 49 in slot 6 to **00:20:da:06:ba:d3** you would enter:

```
-> lacp agg 6/49 partner admin system id 00:20:da:06:ba:d3
```

As an option, you can use the **ethernet**, **fastethernet**, and **gigaethernet** keywords before the slot and port number to document the interface type or make the command look consistent with early-generation Alcatel CLI syntax. For example, to modify the system ID of dynamic aggregate partner port 49 in slot 6 to **00:20:da:06:ba:d3** and document that the port is a Gigabit Ethernet port you would enter:

```
-> lacp agg gigaethernet 6/49 partner admin system id 00:20:da:06:ba:d3
```

Restoring the Partner Port System ID

To remove a user-configured system ID from a dynamic aggregate group partner port's configuration use the **no** form of the **lacp agg partner admin system id** command by entering **lacp agg**, the slot number, a slash (/), the port number, and **no partner admin system id**.

For example, to remove a user-configured system ID from dynamic aggregate partner port 2 in slot 6 you would enter:

```
-> lacp agg 6/2 no partner admin system id
```

Modifying the Partner Port System Priority

By default, the administrative priority of a dynamic aggregate group partner port is 0. The following subsections describe how to configure a user-specified value and how to restore the value to its default value with the **lacp agg partner admin system priority** command.

Configuring the Partner Port System Priority

You can configure the administrative priority of a dynamic aggregate group partner port to a value ranging from 0 to 255 by entering **lacp agg**, the slot number, a slash (/), the port number, **partner admin system priority**, and the user-specified administrative system priority.

For example, to modify the administrative priority of dynamic aggregate partner port 49 in slot 4 to 100 you would enter:

```
-> lacp agg 4/49 partner admin system priority 100
```

As an option, you can use the **ethernet**, **fastethernet**, and **gigaethernet** keywords before the slot and port number to document the interface type or make the command look consistent with early-generation Alcatel CLI syntax. For example, to modify the administrative priority of dynamic aggregate partner port 49 in slot 4 to 100 and specify that the port is a Gigabit Ethernet port you would enter:

```
-> lacp agg gigaethernet 4/49 partner admin system priority 100
```

Restoring the Partner Port System Priority

To remove a user-configured system priority from a dynamic aggregate group partner port's configuration use the **no** form of the **lacp agg partner admin system priority** command by entering **lacp agg**, the slot number, a slash (/), the port number, and **no partner admin system priority**.

For example, to remove a user-configured system ID from dynamic aggregate partner port 3 in slot 4 you would enter:

```
-> lacp agg 4/3 no partner admin system priority
```

Modifying the Partner Port Administrative Status

By default, the administrative status of a dynamic aggregate group partner port is 0. The following subsections describe how to configure a user-specified value and how to restore the value to its default value with the **lacp agg partner admin port** command.

Configuring the Partner Port Administrative Status

You can configure the administrative status of a dynamic aggregate group partner port to a value ranging from 0 to 65535 by entering **lacp agg**, the slot number, a slash (/), the port number, **partner admin port**, and the user-specified partner port administrative status.

For example, to modify the administrative status of dynamic aggregate partner port 1 in slot 7 to 200 you would enter:

```
-> lacp agg 7/1 partner admin port 200
```

As an option, you can use the **ethernet**, **fastethernet**, and **gigaethernet** keywords before the slot and port number to document the interface type or make the command look consistent with early-generation Alcatel CLI syntax. For example, to modify the administrative status of dynamic aggregate partner port 1 in slot 7 to 200 and document that the port is a Fast Ethernet port you would enter:

```
-> lacp agg fastethernet 7/1 partner admin port 200
```

Restoring the Partner Port Administrative Status

To remove a user-configured administrative status from a dynamic aggregate group partner port's configuration use the **no** form of the **lacp agg partner admin port** command by entering **lacp agg**, the slot number, a slash (/), the port number, and **no partner admin port**.

For example, to remove a user-configured administrative status from dynamic aggregate partner port 1 in slot 7 you would enter:

```
-> lacp agg 7/1 no partner admin port
```

Modifying the Partner Port Priority

The default partner port priority is 0. The following subsections describe how to configure a user-specified value and how to restore the value to its default value with the **lacp agg partner admin port priority** command.

Configuring the Partner Port Priority

To configure the partner port priority to a value ranging from 0 to 255 by entering **lacp agg**, the slot number, a slash (/), the port number, **partner admin port priority**, and the user-specified partner port priority.

For example, to modify the port priority of dynamic aggregate partner port 3 in slot 4 to 100 you would enter:

```
-> lacp agg 4/3 partner admin port priority 100
```

As an option, you can use the **ethernet**, **fastethernet**, and **gigaethernet** keywords before the slot and port number to document the interface type or make the command look consistent with early-generation Alcatel CLI syntax. For example, to modify the port priority of dynamic aggregate partner port 3 in slot 4 to 100 and document that the port is a Fast Ethernet port you would enter:

```
-> lacp agg fastethernet 4/3 partner admin port priority 100
```

Restoring the Partner Port Priority

To remove a user-configured partner port priority from a dynamic aggregate group partner port's configuration use the **no** form of the **lacp agg partner admin port priority** command by entering **lacp agg**, the slot number, a slash (/), the port number, and **no partner admin port priority**.

For example, to remove a user-configured partner port priority from dynamic aggregate partner port 3 in slot 4 you would enter:

```
-> lacp agg 4/3 no partner admin port priority
```

Optimizing NI Modules for Link Aggregation

By default, first-generation Network Interface (NI) modules are not optimized for link aggregation. The table below shows which NI modules are first-generation modules and are not optimized for link aggregation. Use the **show ni**, **show module**, or **show ni** command to display the part number of the NI module. Second-generation NI modules are distinguished from first-generation NI modules by “ENI2” or “GNI2” in the part number. (First-generation modules have ENI, GNI, or 10GNI in their part numbers instead.) If the NI is a second-generation module you do not need to optimize it.

To modify the optimization status of an NI module use the **linkagg slot optimization** command. To use this command enter **linkagg slot** followed by the slot number of the NI module and either **enable** or **disable**.

For example, to enable link aggregation optimization on an NI module is Slot 5 enter:

```
-> linkagg slot 5 optimization enable
```

When a port is a member of an aggregate group and optimization is enabled on this NI, all bridged traffic sent from any other port (not part of the aggregate group) on the same switching ASIC to the aggregate will be dropped. In this case, traffic needs to be routed between that port and the aggregate group.

In a chassis with both first-generation and second-generation NI modules you must configure dynamic link aggregation on all of the first-generation NI modules before you *must* configure dynamic link aggregation on all of these NI modules before you configure it on any of the second-generation NI modules. In addition, hot insertion or hot swapping of a first-generation NI module into a chassis that has only second-generation NI modules can cause configuration problems.

Configuring Multiple Link Aggregation Groups on NI Modules

By default, only a single link aggregation group can be configured on a first-generation Network Interface (NI) module if the NI module has been optimized for link aggregation with the **linkagg slot optimization** command. To configure multiple link aggregation groups on these NI modules use the **linkagg slot multiple** command. To use this command enter **linkagg slot** followed by the slot number of the NI module and **multiple**.

For example, to enable multiple link aggregation groups on the NI module in Slot 5 enter:

```
-> linkagg slot 5 multiple
```

To configure a first-generation NI module for a single link aggregation group (i.e., the default) use the **linkagg slot single** command. To use this command enter **linkagg slot** followed by the slot number of the NI module and **single**.

For example, to configure n first-generation NI module in Slot 5 for a single link aggregation group enter:

```
-> linkagg slot 5 single
```

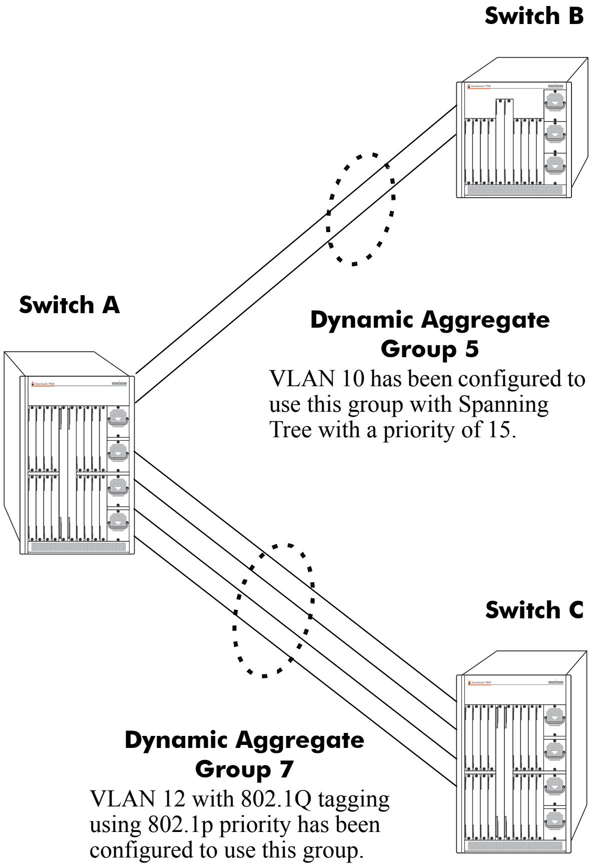
When optimization is enabled on a given NI and multiple link aggregation groups are defined, each aggregate group *must* be part of a different VLAN. In other words, traffic cannot be bridged between aggregate groups on the same NI when optimization is enabled. Instead, traffic must be routed.

Application Examples

Dynamic link aggregation groups are treated by the switch's software the same way it treats individual physical ports.This section demonstrates this feature by providing sample network configurations that use dynamic link aggregation along with other software features. In addition, tutorials are provided that show how to configure these sample networks using Command Line Interface (CLI) commands.

Sample Network Overview

The figure below shows two VLANs on Switch A that use two different link aggregation groups. VLAN 10 has been configured on dynamic aggregate group 5 with Spanning Tree Protocol (STP) with the highest (15) priority possible. And VLAN 12 has been configured on dynamic aggregate group 7 with 802.1Q tagging and 802.1p priority bit settings.


Switch B
Switch A
Dynamic Aggregate Group 5
VLAN 10 has been configured to use this group with Spanning Tree with a priority of 15.
Switch C
Dynamic Aggregate Group 7
VLAN 12 with 802.1Q tagging using 802.1p priority has been configured to use this group.

Sample Network Using Dynamic Link Aggregation

The steps to configure VLAN 10 (Spanning Tree example) are described in "Link Aggregation and Spanning Tree Example" on page 12-31. And the steps to configure VLAN 12 (802.1Q and 802.1p example) are described in "Link Aggregation and QoS Example" on page 12-32.

Note. Although you would need to configure both the local (i.e., Switch A) and remote (i.e., Switches B and C) switches, only the steps to configure the local switch are provided since the steps to configure the remote switches are not significantly different.

Link Aggregation and Spanning Tree Example

As shown in the figure on page 12-30, VLAN 10, which uses the Spanning Tree Protocol (STP) with a priority of 15, has been configured to use dynamic aggregate group 7. The actual physical links connect ports 5/5 and 5/6 on Switch A to ports 4/1 and 4/2 on Switch B. Follow the steps below to configure this network:

Note. Only the steps to configure the local (i.e., Switch A) are provided here since the steps to configure the remote (i.e., Switch B) would not be significantly different.

1 Configure dynamic aggregate group 5 by entering:

```
-> lacp linkagg 5 size 2
```

2 Assign ports 5/5 and 5/6 with same actor administrative key (5) by entering:

```
-> lacp agg 5/5 actor admin key 5
-> lacp agg 5/6 actor admin key 5
```

3 Create VLAN 10 by entering:

```
-> vlan 10
```

4 If the Spanning Tree Protocol (STP) has been disabled on this VLAN (STP is enabled by default), enable it on VLAN 10 by entering:

```
-> vlan 10 stp enable
```

Note. *Optional*. Use the **show spantree ports** command to determine if the STP is enabled or disabled and to display other STP parameters. For example:

```
-> show spantree 10 ports
Spanning Tree Port Summary for Vlan 10
          Adm Oper Man. Path  Desig      Fw  Prim. Adm Op
Port  Pri St  St   mode Cost  Cost  Role Tx  Port  Cnx Cnx Desig Bridge ID
-----+---+---+----+----+-----+-----+----+---+-----+---+---+---------------------
3/13  7   ENA FORW No   100   0     DESG 1   3/13  EDG NPT 000A-00:d0:95:6b:0a:c0
12/10 7   ENA FORW No   19    0     DESG 1   12/10 PTP PTP 000A-00:d0:95:6b:0a:c0
15/2  7   ENA DIS  No   0     0     DIS  0   15/2  EDG NPT 0000-00:00:00:00:00:00
0/5   7   ENA FORW No   4     0     DESG 1   0/10  PTP PTP 000A-00:d0:95:6b:0a:c0
```

In the example above the link aggregation group is indicated by the “0” for the slot number.

5 Configure VLAN 10 (which uses dynamic aggregate group 5) to the highest (15) priority possible by entering:

```
-> bridge 10 5 mode priority 15
```

6 Repeat steps 1 through 5 on Switch B. All the commands would be the same except you would substitute the appropriate port numbers.

Link Aggregation and QoS Example

As shown in the figure on page 12-30, VLAN 12, which uses 802.1Q frame tagging and 802.1p prioritization, has been configured to use dynamic aggregate group 7. The actual physical links connect ports 8/1, 8/2, 8/3, and 8/4 on Switch A to ports 7/1, 7/2, 7/4, and 7/4 on Switch C. Follow the steps below to configure this network:

Note. Only the steps to configure the local (i.e., Switch A) are provided here since the steps to configure the remote (i.e., Switch C) would not be significantly different.

1 Configure dynamic aggregate group 7 by entering:

```
-> lacp linkagg 7 size 4
```

2 Assign ports 8/1, 8/2, 8/3, and 8/4 the same actor administrative key (7) by entering:

```
-> lacp agg 8/1 actor admin key 7
-> lacp agg 8/2 actor admin key 7
-> lacp agg 8/3 actor admin key 7
-> lacp agg 8/4 actor admin key 7
```

3 Create VLAN 12 by entering:

```
-> vlan 12
```

4 Configure 802.1Q tagging with a tagging ID (i.e., VLAN ID) of 12 on dynamic aggregate group 7 by entering:

```
-> vlan 12 802.1q 7
```

5 If the QoS Manager has been disabled (it is enabled by default) enable it by entering:

```
-> qos enable
```

Note. *Optional*. Use the **show qos config** command to determine if the QoS Manager is enabled or disabled.

6 Configure a policy condition for VLAN 12 called "vlan12_condition" by entering:

```
-> policy condition vlan12_condition destination vlan 12
```

7 Configure an 802.1p policy action with the highest priority possible (i.e., 7) for VLAN 12 called "vlan12_action" by entering:

```
-> policy action vlan12_action 802.1P 7
```

8 Configure a QoS rule called "vlan12_rule" using the policy condition and policy rules you configured in steps **8** and **9** above by entering:

```
-> policy rule vlan12_rule enable condition vlan12_condition action
vlan12_action
```

9 Enable your 802.1p QoS settings by entering **qos apply** as shown below:

```
-> qos apply
```

10 Repeat steps 1 through 9 on Switch C. All the commands would be the same except you would substitute the appropriate port numbers.

Note. If you do not use the **qos apply** command any QoS policies you configured will be lost on the next switch reboot.

Displaying Dynamic Link Aggregation Configuration and Statistics

You can use Command Line Interface (CLI) **show** commands to display the current configuration and statistics of link aggregation. These commands include the following:

show linkagg Displays information on link aggregation groups.

show linkagg port Displays information on link aggregation ports.

show linkagg slot optimization Displays the link aggregation optimization status and configuration of an NI module.

When you use the **show linkagg** command without specifying the link aggregation group number and when you use the **show linkagg port** command without specifying the slot and port number, these commands provide a “global” view of switch-wide link aggregate group and link aggregate port information, respectively.

For example, to display global statistics on all link aggregate groups (both dynamic and static) you would enter

```
-> show linkagg
```

A screen similar to the following would be displayed:

```
Number  Aggregate  SNMP Id  Size  Admin State  Oper State     Att/Sel Ports
-------+----------+--------+----+-------------+-------------+-------------
   1      Static    40000001   8     ENABLED      UP              2   2
   2      Dynamic   40000002   4     ENABLED      DOWN            0   0
   3      Dynamic   40000003   8     ENABLED      DOWN            0   2
   4      Dynamic   40000004   16    ENABLED      UP              3   3
   5      Static    40000005   2     DISABLED     DOWN            0   0
```

When you use the **show linkagg** command with the link aggregation group number and when you use the **show linkagg port** command with the slot and port number, these commands provide detailed views of link aggregate group and port information, respectively. These detailed views provide excellent tools for diagnosing and troubleshooting problems.

For example, to display detailed statistics for port 1 in slot 2 that is attached to dynamic link aggregate group 1 you would enter:

```
-> show linkagg port 2/1
```

A screen similar to the following would be displayed:

```
Dynamic Aggregable Port
  SNMP Id                       : 2001,
  Slot/Port                     : 2/1,
  Administrative State          : ENABLED,
  Operational State             : DOWN,
  Port State                    : CONFIGURED,
  Link State                    : DOWN,
  Selected Agg Number           : NONE,
  Primary port                  : UNKNOWN,
LACP
  Actor System Priority         : 10,
  Actor System Id               : [00:d0:95:6a:78:3a],
  Actor Admin Key               : 8,
  Actor Oper Key                : 8,
  Partner Admin System Priority : 20,
  Partner Oper System Priority  : 20,
  Partner Admin System Id       : [00:00:00:00:00:00],
  Partner Oper System Id        : [00:00:00:00:00:00],
  Partner Admin Key             : 8,
  Partner Oper Key              : 0,
  Attached Agg Id               : 0,
  Actor Port                    : 7,
  Actor Port Priority           : 15,
  Partner Admin Port            : 0,
  Partner Oper Port             : 0,
  Partner Admin Port Priority   : 0,
  Partner Oper Port Priority    : 0,
  Actor Admin State             : act1.tim1.agg1.syn0.col0.dis0.def1.exp0
  Actor Oper State              : act1.tim1.agg1.syn0.col0.dis0.def1.exp0,
  Partner Admin State           : act0.tim0.agg1.syn1.col1.dis1.def1.exp0,
  Partner Oper State            : act0.tim0.agg1.syn0.col1.dis1.def1.exp0
```

The **show linkagg slot optimization** command provides the status of link aggregation optimization on a single Network Interface (NI) module. For example, to display the optimization status of the NI module in Slot 5 enter:

```
-> show linkagg slot 5 optimization
```

A screen similar to the following will be displayed:

```
Link aggregation optimization is enabled for slot 5
Multiple aggregates per slot disabled for slot 5
```

Note. See the “Link Aggregation Commands” chapter in the *OmniSwitch CLI Reference Guide* for complete documentation of **show** commands for link aggregation.

---

13 Configuring IP

Internet Protocol (IP) is primarily a network-layer (Layer 3) protocol that contains addressing and control information that enables packets to be forwarded. Along with Transmission Control Protocol (TCP), IP represents the heart of the Internet protocols. IP has two primary responsibilities: providing connectionless, best-effort delivery of datagrams through an internetwork; and providing fragmentation and reassembly of datagrams to support data links with different Maximum Transmission Unit (MTU) sizes.

Note. IP routing (Layer 3) can be accomplished by using static routes or by using one of the IP routing protocols: Routing Information Protocol (RIP), Open Shortest Path First (OSPF), or Border Gateway Protocol (BGP). For more information on these protocols, see Chapter 15, "Configuring RIP," in this manual; or "Configuring OSPF" and "Configuring BGP," in the *OmniSwitch 7700/7800/8800 Advanced Routing Configuration Guide*.

There are two versions of Internet Protocol supported: IPv4 and IPv6. For more information about using IPv6, see Chapter 14, "Configuring IPv6."

In This Chapter

This chapter describes IP and how to configure it through the Command Line Interface (CLI). It includes instructions for enabling IP forwarding, as well as basic IP configuration commands (e.g., **ip router primary-address**). CLI commands are used in the configuration examples; for more details about the syntax of commands, see the *OmniSwitch CLI Reference Guide*.

This chapter provides an overview of IP and includes information about the following procedures:

- IP Forwarding
  - "Configuring an IP Router Interface" on page 13-7
  - "Creating a Static Route" on page 13-10
  - "Creating a Default Route" on page 13-11
  - "Configuring Address Resolution Protocol (ARP)" on page 13-11
- IP Configuration
  - "Configuring the Router Primary Address" on page 13-15
  - "Configuring the Router ID" on page 13-15
  - "Configuring the Time-to-Live (TTL) Value" on page 13-15
  - "IP-Directed Broadcasts" on page 13-16
  - "Denial of Service (DoS) Filtering" on page 13-16

- Managing IP
  - “Internet Control Message Protocol (ICMP)” on page 13-21
  - “Using the Ping Command” on page 13-24
  - “Tracing an IP Route” on page 13-25
  - “Displaying TCP Information” on page 13-25
  - “Displaying UDP Information” on page 13-25

IP Specifications

RFCs Supported	RFC 791–Internet Protocol RFC 792–Internet Control Message Protocol RFC 826–An Ethernet Address Resolution Protocol
Maximum VLANs per switch	4094 (including default VLAN 1)
Maximum router VLANs per switch	4094 IP and 256 IPX (single MAC router mode) 64 IP, IPX or combination of both (multiple MAC router mode)
Maximum IP interfaces per VLAN	8
Maximum IP interfaces per switch	4096
Maximum ARP filters per switch	200

IP Defaults

The following table lists the defaults for IP configuration through the **ip** command.

Description	**Command**	**Default**
IP-Directed Broadcasts	**ip directed-broadcast**	off
Time-to-Live Value	**ip default-ttl**	64 (hops)
IP interfaces	**ip interface**	VLAN 1 interface
ARP filters	**arp filter**	0

Quick Steps for Configuring IP Forwarding

Using only IP, which is always enabled on the switch, devices connected to ports on the same VLAN are able to communicate at Layer 2. The initial configuration for all Alcatel switches consists of a default VLAN 1. All switch ports are initially assigned to this VLAN. When a switching module is added to the switch, all of the module's ports are also assigned to VLAN 1. If additional VLANs are not configured on the switch, the entire switch is treated as one large broadcast domain, and all ports receive all traffic from all other ports.

Note. A VLAN's operational status remains inactive until at least one active switch port is assigned to the VLAN. Ports are considered active if they are connected to an active network device. Non-active port assignments are allowed, but do not change the VLAN's operational state.

To forward packets to a different VLAN on the switch, you must create a router interface on each VLAN. The following steps show you how to enable IP forwarding between VLANs "from scratch". If active VLANs have already been created on the switch, you only need to create router interfaces on each VLAN (Steps 5 and 6).

1 Create VLAN 1 with a description (e.g., VLAN 1) using the **vlan** command. For example:

```
-> vlan 1 name "VLAN 1"
```

2 Create VLAN 2 with a description (e.g., VLAN 2) using the **vlan** command. For example:

```
-> vlan 2 name "VLAN 2"
```

3 Assign an active port to VLAN 1 using the **vlan port default** command. For example, the following command assigns port 1 on slot 1 to VLAN 1:

```
-> vlan 1 port default 1/1
```

4 Assign an active port to VLAN 2 using the **vlan port default** command. For example, the following command assigns port 2 on slot 1 to VLAN 2:

```
-> vlan 2 port default 1/2
```

5 Create an IP router interface on VLAN 1 using the **ip interface** command. For example:

```
-> ip interface vlan-1 address 171.10.1.1 vlan 1
```

6 Create an IP router interface on VLAN 2 using the **ip interface** command. For example:

```
-> ip interface vlan-2 address 171.11.1.1 vlan 2
```

Note. For information about creating a VLAN see Chapter 5, "Configuring VLANs." For information about creating an IP router interface, see "Configuring an IP Router Interface" on page 13-7.

IP Overview

IP is a network-layer (Layer 3) protocol that contains addressing and control information that enables packets to be forwarded on a network. IP is the primary network-layer protocol in the Internet protocol suite. Along with TCP, IP represents the heart of the Internet protocols.

IP Protocols

IP is associated with several Layer 3 and Layer 4 protocols. These protocols are built into the base code loaded on the switch. A brief overview of supported IP protocols is included below.

Transport Protocols

IP is both connectionless (it forwards each datagram separately) and unreliable (it does not guarantee delivery of datagrams). This means that a datagram may be damaged in transit, or thrown away by a busy switch, or simply never make it to its destination. The resolution of these transit problems is to use a Layer 4 transport protocol, such as:

- TCP—A major data transport mechanism that provides reliable, connection-oriented, full-duplex data streams. While the role of TCP is to add reliability to IP, TCP relies upon IP to do the actual delivering of datagrams.
- UDP—A secondary transport-layer protocol that uses IP for delivery. UDP is not connection-oriented and does not provide reliable end-to-end delivery of datagrams. But some applications can safely use UDP to send datagrams that do not require the extra overhead added by TCP. For more information on UDP, see Chapter 17, "Configuring DHCP Relay."

Application-Layer Protocols

Application-layer protocols are used for switch configuration and management:

- Bootstrap Protocol (BOOTP)/Dynamic Host Configuration Protocol (DHCP)—May be used by an end station to obtain an IP address. The switch provides a DHCP Relay that allows BOOTP requests/replies to cross different networks.
- Simple Network Management Protocol (SNMP)—Allows communication between SNMP managers and SNMP agents on an IP network. Network administrators use SNMP to monitor network performance and manage network resources. For more information, see the "Using SNMP" chapter in the *OmniSwitch 7700/7800/8800 Switch Management Guide*.
- Telnet—Used for remote connections to a device. You can telnet to a switch and configure the switch and the network using the CLI.
- File Transfer Protocol (FTP)—Enables the transfer of files between hosts. This protocol is used to load new images onto the switch.

Additional IP Protocols

There are several additional IP-related protocols that may be used with IP forwarding. These protocols are included as part of the base code.

- Address Resolution Protocol (ARP)—Used to match the IP address of a device with its physical (MAC) address. For more information, see "Configuring Address Resolution Protocol (ARP)" on page 13-11.
- Virtual Router Redundancy Protocol (VRRP)—Used to back up routers. For more information, see Chapter 18, "Configuring VRRP."
- Internet Control Message Protocol (ICMP)—Specifies the generation of error messages, test packets, and informational messages related to IP. ICMP supports the **ping** command used to determine if hosts are online. For more information, see "Internet Control Message Protocol (ICMP)" on page 13-21.
- Router Discovery Protocol (RDP)—Used to advertise and discover routers on the LAN. For more information, see Chapter 16, "Configuring RDP."
- Multicast Services—Includes IP multicast switching (IPMS) and IP multicast routing. For more information, see Chapter 26, "Configuring IP Multicast Switching."

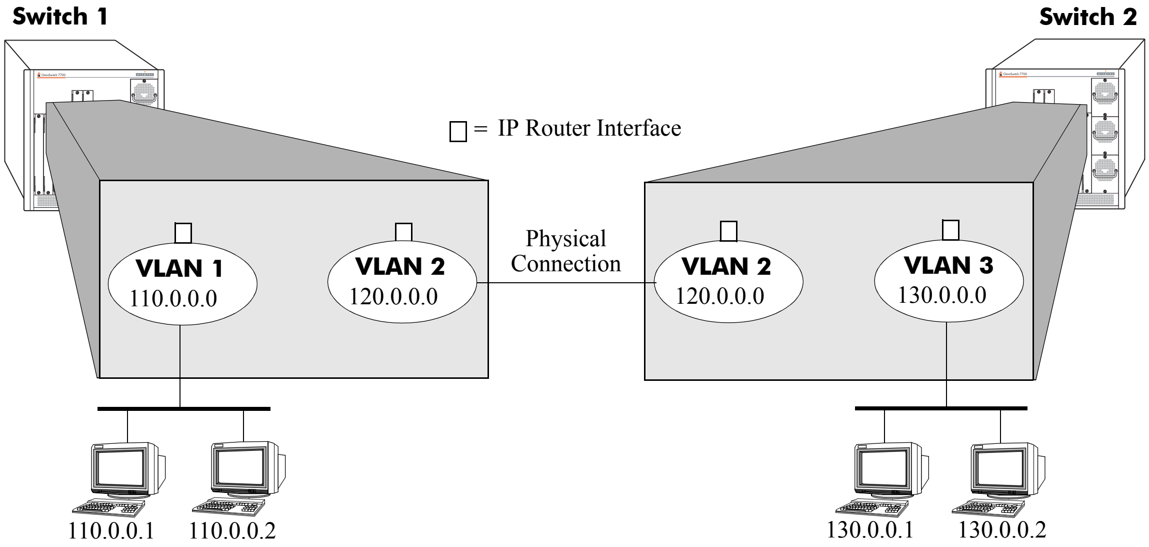
IP Forwarding

Network device traffic is bridged (switched) at the Layer 2 level between ports that are assigned to the same VLAN. However, if a device needs to communicate with another device that belongs to a different VLAN, then Layer 3 routing is necessary to transmit traffic between the VLANs. Bridging makes the decision on where to forward packets based on the packet's destination MAC address; routing makes the decision on where to forward packets based on the packet's IP network address (e.g., IP - 21.0.0.10).

Alcatel switches support routing of IP traffic. A VLAN is available for routing when at least one router interface is defined for that VLAN and at least one active port is associated with the VLAN. If a VLAN does not have a router interface, the ports associated with that VLAN are in essence firewalled from other VLANs.

IP multinetting is also supported. A network is said to be multinetted when multiple IP subnets are brought together within a single broadcast domain. It is now possible to configure up to eight IP interfaces per VLAN and a maximum of 4096 interfaces per switch. Each interface is configured with a different subnet. As a result, traffic from each configured subnet can coexist on the same VLAN.

In the illustration below, an IP router interface has been configured on each VLAN. Therefore, workstations connected to ports on VLAN 1 on Switch 1 can communicate with VLAN 2; and workstations connected to ports on VLAN 3 on Switch 2 can communicate with VLAN 2. Also, ports from both switches have been assigned to VLAN 2, and a physical connection has been made between the switches. Therefore, workstations connected to VLAN 1 on Switch 1 can communicate with workstations connected to VLAN 3 on Switch 2.



IP Forwarding

The number of VLANs that can have IP routing configured depends on which MAC router mode is active on the switch (multiple or single mode). If the switch is running in multiple MAC router mode (the default), then a maximum of 64 VLANs can have IP, IPX, or a combination of both router interfaces defined. In this mode, each VLAN is assigned a unique MAC address. All router interfaces configured for the VLAN share the VLAN MAC address.

Note. Each switch is equipped with 32 MAC addresses. If the multiple MAC router mode is active and more than 32 router VLANs are needed, then additional MAC addresses are required. Contact your Alcatel representative for information about how to increase the number of available MAC addresses.

If the switch is running in single MAC router mode, a maximum of 4094 VLANs can have IP interfaces defined and a maximum of 256 VLANs can have IPX interfaces defined. In this mode, each router VLAN is assigned the same MAC address, which is the base chassis MAC address for the switch.

Regardless of which mode is active on the switch, VLAN 1 router interfaces are always assigned the base chassis MAC address. Use the **show vlan router mac status** command to determine which MAC router mode is active on the switch. See Chapter 5, "Configuring VLANs," for more information about configuring the MAC router mode and IPX router interfaces.

Configuring an IP Router Interface

IP is enabled by default. Using IP, devices connected to ports on the same VLAN are able to communicate. However, to forward packets to a different VLAN, you must create at least one router interface on each VLAN.

Use the **ip interface** command to define up to eight IP interfaces for an existing VLAN. The following parameter values are configured with this command:

- A unique interface name (text string up to 20 characters) used to identify the IP interface. Specifying this parameter is required to create or modify an IP interface.
- The VLAN ID of an existing VLAN.
- An IP address to assign to the router interface (e.g., 193.204.173.21). Note that router interface IP addresses must be unique. You cannot have two router interfaces with the same IP address.
- A subnet mask (defaults to the IP address class).
- The forwarding status for the interface (defaults to forwarding). A forwarding router interface sends IP frames to other subnets. A router interface that is not forwarding can receive frames from other hosts on the same subnet.
- An Ethernet-II or SNAP encapsulation for the interface (defaults to Ethernet-II). The encapsulation determines the framing type the interface uses when generating frames that are forwarded out VLAN ports. Select an encapsulation that matches the encapsulation of the majority of VLAN traffic.
- The Local Proxy ARP status for the VLAN. If enabled, traffic within the VLAN is routed instead of bridged. ARP requests return the MAC address of the IP router interface defined for the VLAN. For more information about Local Proxy ARP, see "Local Proxy ARP" on page 13-13.
- The Maximum Transmission Unit (MTU) packet size for the specified interface (defaults to 1500 bytes). The MTU size is configurable on all VLANs, but only restricts the transmission packet size for VLANs with an IP router interface defined.

- The primary interface status. Designates the specified IP interface as the primary interface for the VLAN. By default, the first interface bound to a VLAN becomes the primary interface for that VLAN.

The following **ip interface** command example creates an IP interface named Marketing with an IP network address of 21.0.0.1 and binds the interface to VLAN 455:

```
-> ip interface Marketing address 21.0.0.1 vlan 455
```

The **name** parameter is the only parameter required with this command. Specifying additional parameters is only necessary to configure a value other than the default value for that parameter. For example, both the following commands will create an IP router interface for VLAN 955 with a class A subnet mask, an enabled forwarding status, Ethernet-II encapsulation, an MTU size of 1500 and a disabled Local Proxy ARP and primary interface status:

```
-> ip interface Accounting address 71.0.0.1 mask 255.0.0.0 vlan 955 forward e2
mtu 1500 no local-proxy-arp no primary
-> ip interface Accounting address 71.0.0.1 vlan 955
```

Note. Assign only ports to the VLAN that are capable of handling the MTU size restrictions configured for the IP interface(s) associated with the VLAN. For example, if an interface MTU size is greater than 1500, do not assign 10/100 Ethernet ports to the VLAN if traffic for the interface will originate or forward on these ports.

Modifying an IP Router Interface

The **ip interface** command is also used to modify existing IP interface parameter values. It is not necessary to first remove the IP interface and then create it again with the new values. The changes specified will overwrite existing parameter values. For example, the following command changes the subnet mask to **255.255.255.0**, the forwarding status to **no forwarding** and the encapsulation to **snap** by overwriting existing parameter values defined for the interface. The interface name, **Accounting**, is specified as part of the command syntax to identify which interface to change.

```
-> ip interface Accounting mask 255.255.255.0 no forward snap
```

Note that when changing the IP address for the interface, the subnet mask will revert back to the default mask value if it was previously set to a non-default value and it is not specified when changing the IP address. For example, the following command changes the IP address for the Accounting interface:

```
-> ip interface Accounting address 40.0.0.1
```

The subnet mask for the Accounting interface was previously set to 255.255.255.0. The above example resets the mask to the default value of 255.0.0.0 because 40.0.0.1 is a Class A address and no other mask was specified with the command. This only occurs when the IP address is modified, all other parameter values remain unchanged unless otherwise specified.

To avoid the problem in the above example, simply enter the non-default mask value whenever the IP address is changed for the interface. For example:

```
-> ip interface Accounting address 40.0.0.1 mask 255.255.255.0
```

Use the **show ip interface** command to verify IP router interface changes. For more information about these commands, see the *OmniSwitch CLI Reference Guide.*

Removing an IP Router Interface

To remove an IP router interface, use the **no** form of the **ip interface** command. Note that it is only necessary to specify the name of the IP interface, as shown in the following example:

```
-> no ip interface Marketing
```

To view a list of IP interfaces configured on the switch, use the **show ip interface** command. For more information about this command, see the *OmniSwitch CLI Reference Guide.*

Configuring a Loopback0 Interface

Loopback0 is the name assigned to an IP interface to identify a consistent address for network management purposes. The Loopback0 interface is not bound to any VLAN, so it will always remain operationally active. This differs from other IP interfaces in that if there are no active ports in the VLAN, all IP interface associated with that VLAN are not active. In addition, the Loopback0 interface provides a unique IP address for the switch that is easily identifiable to network management applications.

This type of interface is created in the same manner as all other IP interfaces, using the **ip interface** command. To identify a Loopback0 interface, enter **Loopback0** for the interface name. For example, the following command creates the Loopback0 interface with an IP address of 10.11.4.1:

```
-> ip interface Loopback0 address 10.11.4.1
```

Note the following when configuring the Loopback0 interface:

- The interface name, “Loopback0”, is case sensitive.
- The **admin** parameter is the only configurable parameter supported with this type of interface.
- The Loopback0 interface is always active and available.
- Only one Loopback0 interface per switch is allowed.
- Creating this interface does *not* deduct from the total number of IP interfaces allowed per VLAN or switch.

Loopback0 Address Advertisement

The Loopback0 IP interface address is automatically advertised by the IGP protocols RIP and OSPF when the interface is created. There is no additional configuration necessary to trigger advertisement with these protocols.

Note the following regarding Loopback0 advertisement:

- RIP advertises the host route to the Loopback0 IP interface as a redistributed (directhost) route.
- OSPF advertises the host route to the Loopback0 IP interface in its Router-LSAs (as a Stub link) as an internal route into all its configured areas.

Configuring a BGP Peer Session with Loopback0

It is possible to create BGP peers using the Loopback0 IP interface address of the peering router and binding the source (i.e., outgoing IP interface for the TCP connection) to its own configured Loopback0 interface. The Loopback0 IP interface address can be used for both Internal and External BGP peer sessions. For EBGP sessions, if the External peer router is multiple hops away, the **ebgp-multihop** parameter may need to be used.

The following example command configures a BGP peering session using a Loopback0 IP interface address:

```
-> ip bgp neighbor 2.2.2.2 update-source Loopback0
```

See the *OmniSwitch 7700/7800/8800 Advanced Routing Configuration Guide* for more information.

Creating a Static Route

Static routes are user-defined and carry a higher priority than routes created by dynamic routing protocols. That is, static routes always have priority over dynamic routes regardless of the metric value. Static routes allow you to define, or customize, an explicit path to an IP network segment, which is then added to the IP Forwarding table. Static routes can be created between VLANs to enable devices on these VLANs to communicate.

Use the **ip static-route** command to create a static route. You must specify the destination IP address of the route as well as the IP address of the first hop (gateway) used to reach the destination. For example, to create a static route to IP address 171.11.0.0 through gateway 171.11.2.1 you would enter:

```
-> ip static-route 171.11.0.0 gateway 171.11.2.1
```

The subnet mask is not required if you want to use the natural subnet mask. By default, the switch imposes a natural mask on the IP address. In the above example, the Class B mask of 255.255.0.0 is implied. If you do not want to use the natural mask, you must enter a subnet mask. For example, to create a static route to IP address 10.255.11.0, you would have to enter the Class C mask of 255.255.255.0:

```
-> ip static-route 10.255.11.0 mask 255.255.255.0 gateway 171.11.2.1
```

When you create a static route, the default metric value of 1 is used. However, you can change the priority of the route by increasing its metric value. The lower the metric value, the higher the priority. This metric is added to the metric cost of the route. The metric range is 1 to 15.

For example:

```
-> ip static-route 10.255.11.0 mask 255.255.255.0 gateway 171.11.2.1 metric 5
```

Static routes do not age out of the IP Forwarding table; you must delete them from the table. Use the **no ip static route** command to delete a static route. You must specify the destination IP address of the route as well as the IP address of the first hop (gateway). For example, to delete a static route to IP address 171.11.0.0 through gateway 171.11.2.1 you would enter:

```
-> no ip static-route 171.11.0.0 gateway 171.11.2.1
```

The IP Forwarding table includes routes learned through one of the routing protocols (RIP, OSPF, BGP) as well as any static routes that are configured. Use the **show ip route** command to display the IP Forwarding table.

Note. A static route is not active unless the gateway it is using is active.

Creating a Default Route

A default route can be configured for packets destined for networks that are unknown to the switch. If routing protocols are disabled, packets can still be forwarded to a router that knows where to send them if a default route is created.

Use the **ip static-route** command to create a default route. You must specify a default route of 0.0.0.0 with a subnet mask of 0.0.0.0, and the IP address of the next hop (gateway). For example, to create a default route through gateway 171.11.2.1 you would enter:

```
-> ip static-route 0.0.0.0 mask 0.0.0.0 gateway 171.11.2.1
```

Note. You cannot create a default route using the EMP port as a gateway.

Configuring Address Resolution Protocol (ARP)

To send packets on a locally connected network, the switch uses ARP to match the IP address of a device with its physical (MAC) address. To send a data packet to a device with which it has not previously communicated, the switch first broadcasts an ARP request packet. The ARP request packet requests the Ethernet hardware address corresponding to an Internet address. All hosts on the receiving Ethernet receive the ARP request, but only the host with the specified IP address responds. If present and functioning, the host with the specified IP address responds with an ARP reply packet containing its hardware address. The switch receives the ARP reply packet, stores the hardware address in its ARP cache for future use, and begins exchanging packets with the receiving device.

The switch stores the hardware address in its ARP cache (ARP table). The table contains a listing of IP addresses and their corresponding translations to MAC addresses. Entries in the table are used to translate 32-bit IP addresses into 48-bit Ethernet or IEEE 802.3 hardware addresses. Dynamic addresses remain in the table until they time out. You can set this timeout value and you can also manually add or delete permanent addresses to/from the table.

Adding a Permanent ARP Entry

As described above, dynamic entries remain in the ARP table for a specified time period before they are automatically removed. However, you can create a permanent entry in the table. Because most hosts support the use of address resolution protocols to determine and cache address information (called dynamic address resolution), you generally do not need to specify permanent ARP entries.

Use the **arp** command to add a permanent entry. You must enter the IP address of the entry followed by its physical (MAC) address. For example to create an entry for IP address 171.11.1.1 with a corresponding physical address of 00:05:02:c0:7f:11, you would enter:

```
-> arp 171.11.1.1 00:05:02:c0:7f:11
```

When you add an entry to the table, the IP address and hardware address (MAC address) are *required*. Optionally, you may also specify:

- **Alias.** Use the **alias** keyword to specify that the switch will act as an alias (proxy) for this IP address. When the alias option is used, the switch responds to all ARP requests for the specified IP address with its own MAC address.

For example:

```
-> arp 171.11.1.1 00:05:02:c0:7f:11 alias
```

Note. The **arp alias** command is not related to proxy ARP as defined in RFC 925. Instead, **arp alias** is similar to the Local Proxy ARP feature, except that it is used to configure the switch as a proxy for only *one* IP address.

Use the **show arp** command to display the ARP table and verify that the entry was added.

Deleting a Permanent Entry from the ARP Table

Permanent entries do not age out of the ARP table. Use the **no arp** command to delete a permanent entry. When deleting an ARP entry, you only need to enter the IP address. For example to delete an entry for IP address 171.11.1.1, you would enter:

```
-> no arp 171.11.1.1
```

Use the **show arp** command to display the table and verify that the entry was deleted.

Note. You can also use the **no arp** command to delete a dynamic entry from the table.

Clearing Dynamic Entries

Dynamic entries can be cleared using the **clear arp-cache** command. This command clears all dynamic entries. Permanent entries must be cleared using the **no arp** command.

Use the **show arp** command to display the table and verify that the table was cleared.

Note. Dynamic entries remain in the ARP table until they time out. If the switch does not receive data from a host for this user-specified time, the entry is removed from the table. If another packet is received from this host, the switch goes through the discovery process again to add the entry to the table. The switch uses the MAC Address table timeout value as the ARP timeout value. Use the **mac-address-table aging-time** command to set the timeout value.

Local Proxy ARP

The Local Proxy ARP feature is an extension of the Proxy ARP feature, but is enabled on an IP interface and applies to the VLAN bound to that interface. When Local Proxy ARP is enabled, all ARP requests received on VLAN member ports are answered with the MAC address of the IP interface that has Local Proxy ARP enabled. In essence, all VLAN traffic is now routed within the VLAN instead of bridged and all ARP requests are blocked between ports in the same VLAN.

This feature is intended for use with port mapping applications where VLANs are one-port connections. This allows hosts on the port mapping device to communicate via the router. ARP packets are still bridged across multiple ports.

Note that Local Proxy ARP takes precedence over any switch-wide Proxy ARP or ARP function. In addition, it is not necessary to configure Proxy ARP in order to use Local Proxy ARP. The two features are independent of each other.

By default, Local Proxy ARP is disabled when an IP interface is created. To enable this feature, use the **ip interface** command. For example:

```
-> ip interface Accounting local-proxy-arp
```

Note that when Local Proxy ARP is enabled for any one IP router interface associated with a VLAN, the feature is applied to the entire VLAN. It is not necessary to enable it for each interface. However, if the IP interface that has this feature enabled is moved to another VLAN, Local Proxy ARP is enabled for the new VLAN and must be enabled on another interface for the old VLAN.

ARP Filtering

ARP filtering is used to determine whether or not the switch responds to ARP requests that contain a specific IP address. This feature is generally used in conjunction with the Local Proxy ARP application; however, ARP filtering is available for use on its own and/or with other applications.

By default, no ARP filters exist in the switch configuration. When there are no filters present, all ARP packets are processed, unless they are blocked or redirected by some other feature.

Use the **arp filter** command to specify the following parameter values required to create an ARP filter:

- An IP address (e.g., 193.204.173.21) used to determine whether or not an ARP packet is filtered.
- An IP mask (e.g. 255.0.0.0) used to identify which part of the ARP packet IP address is compared to the filter IP address.
- An optional VLAN ID to specify that the filter is only applied to ARP packets from that VLAN.
- Which ARP packet IP address to use for filtering (sender or target). If the target IP address in the ARP packet matches a target IP specified in a filter, then the disposition for that filter applies to the ARP packet. If the sender IP address in the ARP packet matches a sender IP specified in a filter, then the disposition for that filter applies to the ARP packet.
- The filter disposition (block or allow). If an ARP packet meets filter criteria, the switch is either blocked from responding to the packet or allowed to respond to the packet depending on the filter disposition. Packets that do not meet any filter criteria are responded to by the switch.

The following **arp filter** command example creates an ARP filter will block the switch from responding to ARP packets that contain a sender IP address that starts with 198:

```
-> arp filter 198.0.0.0 mask 255.0.0.0 sender block
```

Up to 200 ARP filters can be defined on a single switch. To remove an individual filter, use the no form of the **arp filter** command. For example:

```
-> no arp filter 198.0.0.0
```

To clear all ARP filters from the switch configuration, use the **clear arp filter** command. For example:

```
-> clear arp filter
```

Use the **show arp filter** command to verify the ARP filter configuration. For more information about this and other ARP filter commands, see the *OmniSwitch CLI Reference Guide*.

IP Configuration

IP is enabled on the switch by default and there are few options that can, or need to be, configured. This section provides instructions for some basic IP configuration options.

Configuring the Router Primary Address

The router primary address is used by advanced routing protocols (e.g., OSPF) to identify the switch on the network. It is also the address that is used to access the switch for management purposes.

Use the **ip router primary-address** command to configure the router primary address. Enter the command, followed by the IP address. For example, to configure a router primary address of 172.22.2.115, you would enter:

```
-> ip router primary-address 172.22.2.115
```

Note. The router primary IP address is required for BGP.

Configuring the Router ID

By default, the primary address of the router is used as the router ID. However, if a primary address has not been configured, the router ID is used by OSPF to identify the switch on the network. The router ID can be any 32-bit number.

Use the **ip router router-id** command to configure the router ID. Enter the command, followed by the IP address. For example, to configure a router ID of 172.22.2.115, you would enter:

```
-> ip router router-id 172.22.2.115
```

Configuring the Route Preference

By default, the route preference of a router is: local, static, OSPF, RIP, and BGP (highest to lowest).

Use the **ip route-pref** command to change the route preference value of a router. For example, to configure the route preference of an OSPF route, you would enter:

```
-> ip route-pref ospf 15
```

Configuring the Time-to-Live (TTL) Value

The TTL value is the default value inserted into the TTL field of the IP header of datagrams originating from the switch whenever a TTL value is not supplied by the transport layer protocol. The value is measured in hops.

Use the **ip default-ttl** command to set the TTL value. Enter the command, followed by the TTL value. For example, to set a TTL value of 75, you would enter:

```
-> ip default-ttl 75
```

The default hop count is 64. The valid range is 1 to 255. Use the **show ip config** command to display the default TTL value.

IP-Directed Broadcasts

An IP directed broadcast is an IP datagram that has all zeroes or all 1's in the host portion of the destination IP address. The packet is sent to the broadcast address of a subnet to which the sender is not directly attached. Directed broadcasts are used in denial-of-service "smurf" attacks. In a smurf attack, a continuous stream of ping requests is sent from a falsified source address to a directed broadcast address, resulting in a large stream of replies, which can overload the host of the source address. By default, the switch drops directed broadcasts. Typically, directed broadcasts should not be enabled.

Use the **ip directed-broadcast** command to enable or disable IP-directed broadcasts. For example:

```
-> ip directed-broadcast off
```

Use the **show ip config** command to display the IP directed-broadcast state.

Denial of Service (DoS) Filtering

By default, the switch filters denial of service (DoS) attacks, which are security attacks aimed at devices that are available on a private network or the Internet. Some of these attacks aim at system bugs or vulnerability (for example, teardrop attacks), while other types of these types of attacks involve generating large volumes of traffic so that network service will be denied to legitimate network users (such as pepsi attacks). Examples of these attacks include the following:

- ICMP Ping of Death—Ping packets that exceed the largest IP datagram size (65535 bytes) are sent to a host and hang or crash the system.
- SYN Attack—Floods a system with a series of TCP SYN packets, resulting in the host issuing SYN-ACK responses. The half open TCP connections can exhaust TCP resources, such that no other TCP connections are accepted.
- Land Attack—Spoofed packets are sent with the SYN flag set to a host on any open port that is listening. The machine may hang or reboot in an attempt to respond.
- Teardrop/Bonk/Boink attacks—Bonk/boink/teardrop attacks generate IP fragments in a special way to exploit IP stack vulnerabilities. If the fragments overlap the way those attacks generate packets, an attack is recorded. Since teardrop, bonk and boink all use the same IP fragmentation mechanism to attack, these is no distinction between detection of these attacks. The old IP fragments in the fragmentation queue is also reaped once the reassemble queue goes above certain size.
- Pepsi Attack—The most common form of UDP flooding directed at harming networks. A pepsi attack is an attack consisting of a large number of spoofed UDP packets aimed at diagnostic ports on network devices. This can cause network devices to use up a large amount of CPU time responding to these packets.

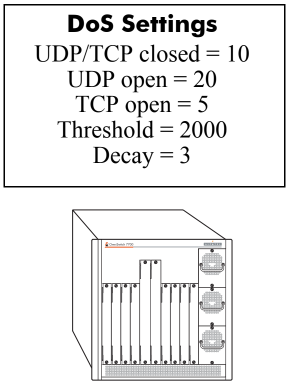
The switch can be set to detect various types of port scans by monitoring for TCP or UDP packets sent to open or closed ports. Monitoring is done in the following manner:

- **Packet penalty values set**. TCP and UDP packets destined for open or closed ports are assigned a penalty value. Each time a packet of this type is received, its assigned penalty value is added to a running total. This total is cumulative and includes all TCP and UDP packets destined for open or closed ports.
- **Port scan penalty value threshold**.The switch is given a port scan penalty value threshold. This number is the maximum value the running penalty total can achieve before triggering an SNMP trap.
- **Decay value**. A decay value is set. The running penalty total is divided by the decay value every minute.

- **Trap generation**. If the total penalty value exceeds the set port scan penalty value threshold, a trap is generated to alert the administrator that a port scan may be in progress.

For example, imagine that a switch is set so that TCP and UDP packets destined for closed ports are given a penalty of 10, TCP packets destined for open ports are given a penalty of 5, and UDP packets destined for open ports are given a penalty of 20. The decay is set to 2, and the switch port scan penalty value threshold is set to 2000:

.

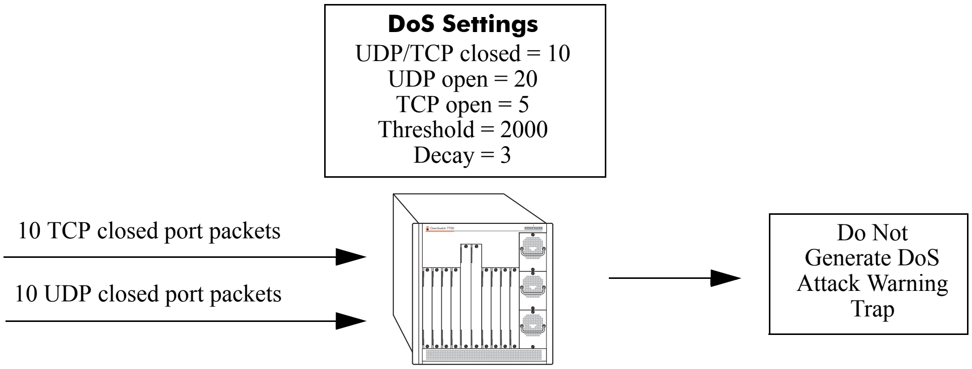


Penalty Total = 0

In one minute, 10 TCP closed port packets and 10 UDP closed port packets are received. This would bring the total penalty value to 200, as shown with the following equation:

```
(10 TCP X 10 penalty) + (10 UDP X 10 penalty) = 200
```

This value would be divided by 2 (due to the decay) and decreased to 100. The switch would not record a port scan:

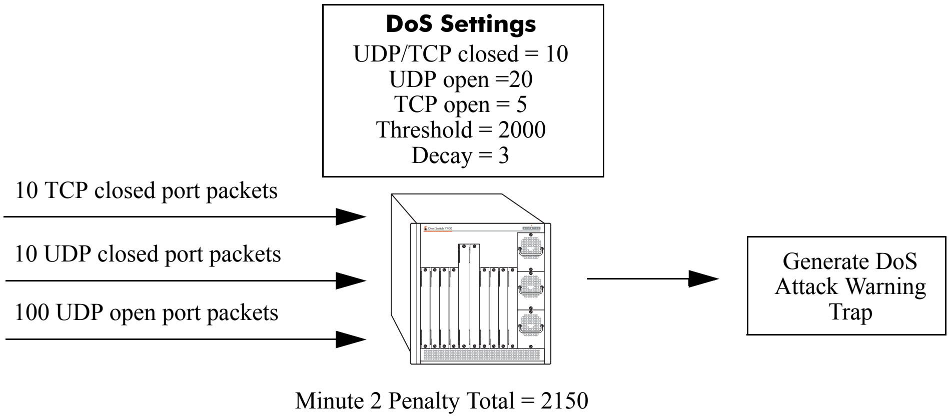

DoS Settings
UDP/TCP closed = 10
UDP open = 20
TCP open = 5
Threshold = 2000
Decay = 3
10 TCP closed port packets
10 UDP closed port packets
Do Not Generate DoS Attack Warning Trap

Minute 1 Penalty Total = 100

In the next minute, 10 more TCP and UDP closed port packets are received, along with 200 UDP open port packets. This would bring the total penalty value to 4300, as shown with the following equation:

```
(100 previous minute value) + (10 TCP X 10 penalty) + (10 UDP X 10 penalty) +
(200 UDP X 20 penalty) = 4300
```

This value would be divided by 2 (due to decay) and decreased to 2150. The switch would record a port scan and generate a trap to warn the administrator:


DoS Settings
UDP/TCP closed = 10
UDP open =20
TCP open = 5
Threshold = 2000
Decay = 3
10 TCP closed port packets
10 UDP closed port packets
100 UDP open port packets
Generate DoS Attack Warning Trap
Minute 2 Penalty Total = 2150

The above functions and how to set their values are covered in the sections that follow.

Setting Penalty Values

There are three types of traffic you can set a penalty value for:

- TCP/UDP packets bound for closed ports.
- TCP traffic bound for open ports.
- UDP traffic bound for open ports.

Each type has its own command used to assign a penalty value. Penalty values can be any non-negative integer. Each time a packet is received that matches an assigned penalty, the total penalty value for the switch is increased by the penalty value of the packet in question.

To assign a penalty value to TCP/UDP packets bound for a closed port, use the **ip dos scan close-port-penalty** command with a penalty value. For example, to assign a penalty value of 10 to TCP/UDP packets destined for closed ports, enter the following:

```
-> ip dos scan close-port-penalty 10
```

To assign a penalty value to TCP packets bound for an open port, use the **ip dos scan tcp open-port-penalty** command with a penalty value. For example, to assign a penalty value of 10 to TCP packets destined for opened ports, enter the following:

```
-> ip dos scan tcp open-port-penalty 10
```

To assign a penalty value to UDP packets bound for an open port, use the **ip dos scan udp open-port-penalty** command with a penalty value. For example, to assign a penalty value of 10 to TCP/UDP packets destined for closed ports, enter the following:

```
-> ip dos scan udp open-port-penalty 10
```

Setting the Port Scan Penalty Value Threshold

The port scan penalty value threshold is the highest point the total penalty value for the switch can reach before a trap is generated informing the administrator that a port scan is in progress.

To set the port scan penalty value threshold, enter the threshold value with the **ip dos scan threshold** command. For example, to set the port scan penalty value threshold to 2000, enter the following:

```
-> ip dos scan threshold 2000
```

Setting the Decay Value

The decay value is the amount the total penalty value is divided by every minute. As the switch records incoming UDP and TCP packets, it adds their assigned penalty values together to create the total penalty value for the switch. To prevent the switch from registering a port scan from normal traffic, the decay value is set to lower the total penalty value every minute to compensate from normal traffic flow.

To set the decay value, enter the decay value with the **ip dos scan decay** command. For example, to set the decay value to 2, enter the following:

```
-> ip dos scan decay 2
```

Enabling DoS Traps

DoS traps must be enabled in order for the switch to warn the administrator that a port scan may be in progress when the switch total penalty value crosses the port scan penalty value threshold.

To enable SNMP trap generation, enter the **ip dos trap** command, as shown:

```
-> ip dos trap enable
```

To disable DoS traps, enter the same **ip dos trap** command, as shown:

```
-> ip dos trap disable
```

Enabling/Disabling IP Services

When a switch initially boots up, all supported TCP/UDP well-known service ports are enabled (open). Although these ports provide access for essential switch management services, such as telnet, ftp, snmp, etc., they also are vulnerable to DoS attacks. It is possible to scan open service ports and launch such attacks based on well-known port information.

The **ip service** command allows you to selectively disable (close) TCP/UDP well-known service ports and enable them when necessary. This command only operates on TCP/UDP ports that are opened by default. It has no effect on ports that are opened by loading applications, such as RIP, BGP, etc.

In addition, the **ip service** command allows you to designate which port to enable or disable by specifying the name of a service or the well-known port number associated with that service. For example, both of the following commands disable the telnet service:

```
-> no ip service telnet
-> no ip service port 23
```

Note that specifying a port number requires the use of the optional **port** keyword.

To enable or disable more than one service in a single command line, enter each service name separated by a space. For example, the following command enables the telnet, ftp, and snmp service ports:

```
-> ip service telnet ftp snmp
```

The following table lists **ip service** command options for specifying TCP/UDP services and also includes the well-known port number associated with each service:

service	port
ftp	21
ssh	22
telnet	23
http	80
secure-http	443
avlan-http	260
avlan-secure-http	261
avlan-telnet	259
udp-relay	67
network-time	123
snmp	161
proprietary	1024
proprietary	1025

Managing IP

The following sections describe IP commands that can be used to monitor and troubleshoot IP forwarding on the switch.

Internet Control Message Protocol (ICMP)

ICMP is a network layer protocol within the IP protocol suite that provides message packets to report errors and other IP packet processing information back to the source. ICMP generates several kinds of useful messages, including Destination Unreachable, Echo Request and Reply, Redirect, Time Exceeded, and Router Advertisement and Solicitation. If an ICMP message cannot be delivered, a second one is not generated. This prevents an endless flood of ICMP messages.

When an ICMP destination-unreachable message is sent by a switch, it means that the switch is unable to send the package to its final destination. The switch then discards the original packet. There are two reasons why a destination might be unreachable. Most commonly, the source host has specified a non-existent address. Less frequently, the switch does not have a route to the destination. Destination-unreachable messages include four basic types:

- Network-Unreachable Message—Usually means that a failure has occurred in the route lookup of the destination IP in the packet.
- Host-Unreachable Message—Usually indicates delivery failure, such as an unresolved client's hardware address or an incorrect subnet mask.
- Protocol-Unreachable Message—Usually means that the destination does not support the upper-layer protocol specified in the packet.
- Port-Unreachable Message—Implies that the TCP/UDP socket or port is not available.

Additional ICMP messages include:

- Echo-Request Message—Generated by the ping command, the message is sent by any host to test node reachability across an internetwork. The ICMP echo-reply message indicates that the node can be successfully reached.
- Redirect Message—Sent by the switch to the source host to stimulate more efficient routing. The switch still forwards the original packet to the destination. ICMP redirect messages allow host routing tables to remain small because it is necessary to know the address of only one switch, even if that switch does not provide the best path. Even after receiving an ICMP redirect message, some devices might continue using the less-efficient route. It is important to note that ICMP redirect messages will never override routing protocols. They are only relevant if routing is not enabled.
- Time-Exceeded Message—Sent by the switch if an IP packet’s TTL field reaches zero. The TTL field prevents packets from continuously circulating the internetwork if the internetwork contains a routing loop. Once a packet’s TTL field reaches 0, the switch discards the packet.

Activating ICMP Control Messages

ICMP messages are identified by a *type* and a *code*. This number pair specifies an ICMP message. For example, ICMP type 4, code 0, specifies the source quench ICMP message.

To enable or disable an ICMP message, use the **icmp type** command with the type and code. For example, to enable the source quench ICMP message (type 4, code 0) enter the following:

```
-> icmp type 4 code 0 enable
```

The following table is provided to identify the various ICMP messages, and their type and code:

ICMP Message	Type	Code
echo reply	**0**	**0**
network unreachable	**0**	**3**
host unreachable	**3**	**1**
protocal unreachable	**3**	**2**
port unreachable	**3**	**3**
frag needed but DF bit set	**3**	**4**
source route failed	**3**	**5**
destination network unknown	**3**	**6**
destination host unknown	**3**	**7**
source host isolated	**3**	**8**
dest network admin prohibited	**3**	**9**
host admin prohibited by filter	**3**	**10**
network unreachable for TOS	**3**	**11**
host unreachable for TOS	**3**	**12**
source quench	**4**	**0**
redirect for network	**5**	**0**
redirect for host	**5**	**1**
redirect for TOS and network	**5**	**2**
redirect for TOS and host	**5**	**3**
echo request	**8**	**0**
router advertisement	**9**	**0**
router solicitation	**10**	**0**
time exceeded during transmit	**11**	**0**
time exceeded during reassembly	**11**	**1**
ip header bad	**12**	**0**
required option missing	**12**	**1**
timestamp request	**13**	**0**
timestamp reply	**14**	**0**
information request (obsolete)	**15**	**0**
information reply (obsolete)	**16**	**0**
address mask request	**17**	**0**
address mask reply	**18**	**0**

In addition to the **icmp type** command, several commonly used ICMP messages have separate CLI commands for convenience. These commands are listed below with the ICMP message name, type, and code:

ICMP Message	Command
Network unreachable (type 0, code 3)	icmp unreachable
Host unreachable (type 3, code 1)	icmp unreachable
Protocol unreachable (type 3, code 2)	icmp unreachable
Port unreachable (type 3, code 3)	icmp unreachable
Echo reply (type 0, code 0)	icmp echo
Echo request (type 8, code 0)	icmp echo
Timestamp request (type 13, code 0)	icmp timestamp
Timestamp reply (type 14, code 0)	icmp timestamp
Address Mask request (type 17, code 0)	icmp addr-mask
Address Mask reply (type 18, code 0)	icmp addr-mask

These commands are entered as the **icmp type** command, only without specifying a type or code. The echo, timestamp, and address mask commands have options for distinguishing between a request or a reply, and the unreachable command has options distinguishing between a network, host, protocol, or port.

For example, to enable an echo request message, enter the following:

```
-> icmp echo request enable
```

To enable a network unreachable message, enter the following:

```
-> icmp unreachable net-unreachable enable
```

See Chapter 1, “IP Commands,” for specifics on the ICMP message commands.

Enabling All ICMP Types

To enable all ICMP message types, use the **icmp messages** command with the **enable** keyword. For example:

```
-> icmp messages enable
```

To disable all ICMP messages, enter the same command with the **disable** keyword. For example:

```
-> icmp messages enable
```

Setting the Minimum Packet Gap

The minimum packet gap is the time required between sending messages of a like type. For instance, if the minimum packet gap for Address Mask request messages is 40 microseconds, and an Address Mask message is sent, at least 40 microseconds must pass before another one could be sent.

To set the minimum packet gap, use the **min-pkt-gap** keyword with any of the ICMP control commands. For example, to set the Source Quench minimum packet gap to 100 microseconds, enter the following:

```
-> icmp type 4 code 0 min-pkt-gap 100
```

Likewise, to set the Timestamp Reply minimum packet gap to 100 microseconds, enter the following:

```
-> icmp timestamp reply min-pkt-gap 100
```

The default minimum packet gap for ICMP messages is 0.

ICMP Control Table

The ICMP Control table displays ICMP control messages, whether they are enabled or disabled, and the minimum packet gap times. Use the **show icmp control** command to display the table.

ICMP Statistics Table

The ICMP Statistics table displays ICMP statistics and errors. This data can be used to monitor and troubleshoot IP on the switch. Use the **show icmp statistics** command to display the table.

Using the Ping Command

The **ping** command is used to test whether an IP destination can be reached from the local switch. This command sends an ICMP echo request to a destination and then waits for a reply. To ping a destination, enter the **ping** command and enter either the destination's IP address or host name. The switch will ping the destination using the default frame count, packet size, interval, and timeout parameters (6 frames, 64 bytes, 1 second, and 5 seconds, respectively). For example:

```
-> ping 172.22.2.115
```

When you ping a device, the device IP address or host name are required. Optionally, you may also specify:

- **Count.** Use the **count** keyword to set the number of frames to be transmitted.
- **Size.** Use the **size** keyword to set the size, in bytes, of the data portion of the packet sent for this ping. You can specify a size or a range of sizes up to 60000.
- **Interval.** Use the **interval** keyword to set the frequency, in seconds, that the switch will poll the host.
- **Timeout.** Use the timeout keyword to set the number of seconds the program will wait for a response before timing out.

For example, to send a ping with a count of 2, a size of 32 bytes, an interval of 2 seconds, and a timeout of 10 seconds you would enter:

```
-> ping 172.22.2.115 count 2 size 32 interval 2 timeout 10
```

Note. If you change the default values they will only apply to the current ping. The next time you use the **ping** command, the default values will be used unless you again enter different values.

Tracing an IP Route

The **traceroute** command is used to find the path taken by an IP packet from the local switch to a specified destination. This command displays the individual hops to the destination as well as some timing information. When using this command, you must enter the name of the destination as part of the command line (either the IP address or host name). Use the optional **max-hop** parameter to set a maximum hop-count to the destination. If the trace reaches this maximum hop-count without reaching the destination, the trace stops.

For example, to perform a tracerout to a device with an IP address of 172.22.2.115 with a maximum hop-count of 10 you would enter:

```
-> traceroute 172.22.2.115 max-hop 10
```

Displaying TCP Information

Use the **show tcp statistics** command to display TCP statistics. Use the **show tcp ports** command to display TCP port information.

Displaying UDP Information

UDP is a secondary transport-layer protocol that uses IP for delivery. UDP is not connection-oriented and does not provide reliable end-to-end delivery of datagrams. But some applications can safely use UDP to send datagrams that do not require the extra overhead added by TCP. Use the **show udp statistics** command to display UDP statistics. Use the **show udp ports** command to display UDP port information.

Verifying the IP Configuration

A summary of the show commands used for verifying the IP configuration is given here:

show ip interface	Displays the usability status of interfaces configured for IP.
show ip route	Displays the IP Forwarding table.
show ip route-pref	Displays a list of all routes (static and dynamic) that exist in the IP router database.
show ip emp-route	Displays the IP routes associated with the Ethernet Management Port (EMP).
show ip config	Displays IP configuration parameters.
show ip protocols	Displays switch routing protocol information and status.
show ip service	Displays the current status of TCP/UDP service ports. Includes service name and well-known port number.
show arp	Displays the ARP table.
show arp filter	Displays the ARP filter configuration for the switch.
show icmp control	This command allows the viewing of the ICMP control settings.
show ip dos config	Displays the configuration parameters of the DoS scan for the switch.
show ip dos statistics	Displays the statistics on detected port scans for the switch.

For more information about the displays that result from these commands, see the *OmniSwitch CLI Reference Guide*.

14 Configuring IPv6

Internet Protocol version 6 (IPv6) is the next generation of Internet Protocol version 4 (IPv4). Both versions are supported along with the ability to tunnel IPv6 traffic over IPv4. Implementing IPv6 solves the limited address problem currently facing IPv4, which provides a 32-bit address space. IPv6 increases the address space available to 128 bits.

Note. On OmniSwitch 6600/7700/7800/8800 series switches, IPv6 is a software based implementation.

In This Chapter

This chapter describes IPv6 and how to configure it through the Command Line Interface (CLI). CLI commands are used in the configuration examples; for more details about the syntax of commands, see the *OmniSwitch CLI Reference Guide*.

This chapter provides an overview of IPv6 and includes information about the following procedures:

- "Configuring an IPv6 Interface" on page 14-10.
- "Assigning IPv6 Addresses" on page 14-12.
- "Configuring IPv6 Tunnel Interfaces" on page 14-14.

IPv6 Specifications

RFCs Supported	2460–*Internet Protocol, Version 6 (IPv6) Specification* 2461–*Neighbor Discovery for IP Version 6 (IPv6)* 2462–*IPv6 Stateless Address Autoconfiguration* 2463–*Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification* 2464–*Transmission of IPv6 Packets Over Ethernet Networks* 2893–*Transition Mechanisms for IPv6 Hosts and Routers* 3513–*Internet Protocol Version 6 (IPv6) Addressing Architecture* 3056–*Connection of IPv6 Domains via IPv4 Clouds*
Maximum VLANs per switch	4094 (including default VLAN 1)
Maximum IPv6 router VLANs per switch	4094 IP and 256 IPX (single MAC router mode) 64 IP, IPX or combination of both (multiple MAC router mode)
Maximum IPv4 router VLANs per switch	4094 IP and 256 IPX (single MAC router mode) 64 IP, IPX or combination of both (multiple MAC router mode)
Maximum IPv6 interfaces per VLAN	1
Maximum IPv6 interfaces per tunnel	1
Maximum 6to4 tunnels per switch	1
Maximum configured tunnels per switch	255

IPv6 Defaults

The following table lists the defaults for IPv6 configuration through the **ip** command.

Description	Command	Default
Global status of IPv6 on the switch	**N/A**	Enabled
IPv6 interfaces	**ipv6 interface**	None

Quick Steps for Configuring IPv6 Routing

The following tutorial assumes that VLAN 200 and VLAN 300 already exist in the switch configuration. For information about how to configure VLANs, see Chapter 5, “Configuring VLANs.”

1 Configure an IPv6 interface for VLAN 200 using the **ipv6 interface** command. For example:

```
-> ipv6 interface v6if-v200 vlan 200
```

Note that when the IPv6 interface is configured, the switch automatically generates a link-local address for the interface. This allows for communication with other interfaces and/or devices on the same link, but does not provide routing between interfaces.

2 Assign a unicast address to the *v6if-v200* interface using the **ipv6 address** command. For example:

```
-> ipv6 address 4100:1::/64 eui-64 v6if-v200
```

3 Configure an IPv6 interface for VLAN 300 using the **ipv6 interface** command. For example:

```
-> ipv6 interface v6if-v300 vlan 300
```

4 Assign a unicast address to the *v6if-v300* interface using the **ipv6 address** command. For example:

```
-> ipv6 address 4100:2::/64 eui-64 v6if-v300
```

Note. *Optional*. To verify the IPv6 interface configuration, enter **show ipv6 interface** For example:

```
-> show ipv6 interface
Name                 IPv6 Address/Prefix Length                 Status  Device
--------------------+------------------------------------------+-------+-----------
v6if-v200            fe80::2d0:95ff:fe12:fab5/64                 Down    VLAN 200
                     4100:1::2d0:95ff:fe12:fab5/64
                     4100:1::/64
v6if-v300            fe80::2d0:95ff:fe12:fab6/64                 Down    VLAN 300
                     4100:2::2d0:95ff:fe12:fab6/64
                     4100:2::/64
loopback             ::1/128                                     Active  Loopback
                     fe80::1/64
```

Note that the link-local addresses for the two new interfaces and the loopback interface were automatically created and included in the **show ipv6 interface** display output. In addition, the subnet router anycast address that corresponds to the unicast address is also automatically generated for the interface

5 Enable RIPng for the switch using the **ipv6 load rip** command. For example:

```
-> ipv6 load rip
```

6 Create a RIPng interface for each of the IPv6 VLAN interfaces using the **ipv6 rip interface** command. For example:

```
-> ipv6 rip interface v6if-v200
-> ipv6 rip interface v6if-v300
```

IPv6 routing is now configured for VLAN 200 and VLAN 300 interfaces, but is not active until at least one port in each VLAN goes active.

IPv6 Overview

IPv6 provides the basic functionality that is offered with IPv4 but includes the following enhancements and features not available with IPv4:

- **Increased IP address size**—IPv6 uses a 128-bit address, a substantial increase over the 32-bit IPv4 address size. Providing a larger address size also significantly increases the address space available, thus eliminating the concern over running out of IP addresses. See "IPv6 Addressing" on page 14-5 for more information.
- **Autoconfiguration of addresses**—When an IPv6 interface is created or a device is connected to the switch, an IPv6 link-local address is automatically assigned for the interface and/or device. See "Autoconfiguration of IPv6 Addresses" on page 14-6 for more information.
- **Anycast addresses**—A new type of address. Packets sent to an anycast address are delivered to one member of the anycast group.
- **Simplified header format**—A simpler IPv6 header format is used to keep the processing and bandwidth cost of IPv6 packets as low as possible. As a result, the IPv6 header is only twice the size of the IPv4 header despite the significant increase in address size.
- **Improved support for header options**—Improved header option encoding allows more efficient forwarding, fewer restrictions on the length of options, and greater flexibility to introduce new options.
- **Security improvements**—Extension definitions provide support for authentication, data integrity, and confidentiality.
- **Neighbor Discovery protocol**—A protocol defined for IPv6 that detects neighboring devices on the same link and the availability of those devices. Additional information that is useful for facilitating the interaction between devices on the same link is also detected (e.g., neighboring address prefixes, address resolution, duplicate address detection, link MTU and hop limit values, etc.).

This implementation of IPv6 also provides the following mechanisms to maintain compatibility between IPv4 and IPv6:

- Dual-stack support for both IPv4 and IPv6 on the same switch.
- Configuration of IPv6 and IPv4 interfaces on the same VLAN.
- Tunneling of IPv6 traffic over an IPv4 network infrastructure.
- Embedded IPv4 addresses in the four lower-order bits of the IPv6 address.

The remainder of this section provides a brief overview of the new IPv6 address notation, autoconfiguration of addresses, and tunneling of IPv6 over IPv4.

IPv6 Addressing

One of the main differences between IPv6 and IPv4 is that the address size increased from 32 bits to 128 bits. Going to a 128-bit address also increases the size of the address space to the point where running out of IPv6 addresses is not a concern.

The following types of IPv6 addresses are supported:

Unicast—Standard unicast addresses, similar to IPv4.

Multicast—Addresses that represent a group of devices. Traffic sent to a multicast address is delivered to all members of the multicast group.

Anycast—Traffic that is sent to this type of address is delivered to one member of the anycast group. The device that receives the traffic is usually the one that is easiest to reach as determined by the active routing protocol.

Note. IPv6 does not support the use of broadcast addresses. This functionality is replaced using improved multicast addressing capabilities.

IPv6 address types are identified by the high-order bits of the address, as shown in the following table:

Address Type	Binary Prefix	IPv6 Notation
Unspecified	00...0 (128 bits)	::/128
Loopback	00...1 (128 bits)	::1/128
Multicast	11111111	FF00::/8
Link-local unicast	1111111010	FE80::/10
Site-local unicast	1111111011	FEC0::/10
Global unicast	everything else	

Note that anycast addresses are unicast addresses that are not identifiable by a known prefix.

IPv6 Address Notation

IPv4 addresses are expressed using dotted decimal notation and consist of four eight-bit octets. If this same method was used for IPv6 addresses, the address would contain 16 such octets, thus making it difficult to manage. IPv6 addresses are expressed using *colon hexidecimal notation* and consist of eight 16-bit words, as shown in the following example:

1234:000F:531F:4567:0000:0000:BCD2:F34A

Note that any field may contain all zeros or all ones. In addition, it is possible to shorten IPv6 addresses by suppressing leading zeros. For example:

1234:F:531F:4567:0:0:BCD2:F34A

Another method for shortening IPv6 addresses, is known as *zero compression*. When an address contains contiguous words that consist of all zeros, a double colon (::) is used to identify these words. For example, using zero compression the address 0:0:0:0:1234:531F:BCD2:F34A is expressed as follows:

::1234:531F:BCD2:F34A

Because the last four words of the above address are uncompressed values, the double colon indicates that the first four words of the address all contain zeros. Note that using the double colon is only allowed once within a single address. So if the address was1234:531F:0:0:BCD2:F34A:0:0, a double colon could *not* replace both sets of zeros. For example, the first two versions of this address shown below are valid, the last version is not valid:

1 1234:531F::BCD2:F34A:0:0

2 1234:531F:0:0:BCD2:F34A::

3 1234:531F::BCD2:F34A:: (not valid)

With IPv6 addresses that have long strings of zeros, the benefit of zero compression is more dramatic. For example, address FF00:0:0:0:0:0:4501:32 becomes FF00::4501:32.

Note that hexidecimal notation used for IPv6 addresses resembles that which is used for MAC addresses. However, it is important to remember that IPv6 addresses still identify a device at the Layer 3 level and MAC addresses identify a device at the Layer 2 level.

Another supported IPv6 address notation includes embedding an IPv4 address as the four lower-order bits of the IPv6 address. This is especially useful when dealing with a mixed IPv4/IPv6 network. For example:

0:0:0:0:0:0:212.100.13.6

IPv6 Address Prefix Notation

The Classless Inter-Domain Routing (CIDR) notation is used to express IPv6 address prefixes. This notation consists of the 128-bit IPv6 address followed by a slash (/) and a number representing the prefix length (IPv6-address/prefix-length). For example, the following IPv6 address has a prefix length of 64 bits:

FE80::2D0:95FF:FE12:FAB2/64

Autoconfiguration of IPv6 Addresses

This implementation of IPv6 supports the *stateless* autoconfiguration of link-local addresses for IPv6 VLAN and tunnel interfaces and for devices when they are connected to the switch. Stateless refers to the fact that little or no configuration is required to generate such addresses and there is no dependency on an address configuration server, such as a DHCP server, to provide the addresses.

A link-local address is a private unicast address that identifies an interface or device on the local network. This type of address allows communication with devices and/or neighboring nodes that are attached to the same physical link. Routing between link-local addresses is not available, link-local addresses are not known or advertised to the general network.

When an IPv6 VLAN or tunnel interface is created or a device is connected to the switch, a link-local address is automatically generated for the interface or device. This type of address consists of the well-known IPv6 prefix FE80::/64 combined with an interface ID. The interface ID is derived from the router MAC address associated with the IPv6 interface or the source MAC address if the address is for a device. The resulting link-local address resembles the following example:

FE80::2d0:95ff:fe6b:5ccd/64

Note that when this example address was created, the MAC address was modified by complementing the second bit of the leftmost byte and by inserting the hex values 0xFF and 0xFE between the third and fourth octets of the address. These modifications were done because IPv6 requires an interface ID that is derived using Modified EUI-64 format.

Stateless autoconfiguration is not available for assigning a global unicast or anycast address to an IPv6 interface. In other words, manual configuration is required to assign a non-link-local address to an interface. See "Assigning IPv6 Addresses" on page 14-12 for more information.

Both stateless and *stateful* autoconfiguration is supported for devices, such as a workstation, when they are connected to the switch. When the stateless method is used in this instance, the device listens for router advertisements to obtain a subnet prefix. The unicast address for the device is then formed by combining the subnet prefix with the interface ID for that device.

Stateful autoconfiguration refers to the use of an independent server, such as a DHCP server, to obtain an IPv6 unicast address and other related information. Of course, manual configuration of an IPv6 address is always available for devices as well.

Regardless of how an IPv6 address is obtained, duplicate address detection (DAD) is performed before the address is assigned to an interface or device. If a duplicate is found, the address is not assigned. Note that DAD is *not* performed for anycast addresses.

Please refer to RFCs 2462, 2464, and 3513 for more technical information about autoconfiguration and IPv6 address notation.

Tunneling IPv6 over IPv4

It is likely that IPv6 and IPv4 network infrastructures will coexist for some time, if not indefinitely. Tunneling provides a mechanism for transitioning an IPv4 network to IPv6 and/or maintaining interoperability between IPv4 and IPv6 networks. This implementation of IPv6 supports tunneling of IPv6 traffic over IPv4. There are two types of tunnels supported: *6to4* and *configured*.

Note. RIPng is not supported over 6to4 tunnels. However, it is possible to create a RIPng interface for a configured tunnel. See "Configuring IPv6 Tunnel Interfaces" on page 14-14 for more information.

6to4 Tunnels

6to4 tunneling provides a mechanism for transporting IPv6 host traffic over an IPv4 network infrastructure to other IPv6 hosts and/or domains without having to configure explicit tunnel endpoints. Instead, an IPv6 6to4 tunnel interface is created at points in the network where IPv6 packets are encapsulated (IPv4 header added) prior to transmission over the IPv4 network or decapsulated (IPv4 header stripped) for transmission to an IPv6 destination.

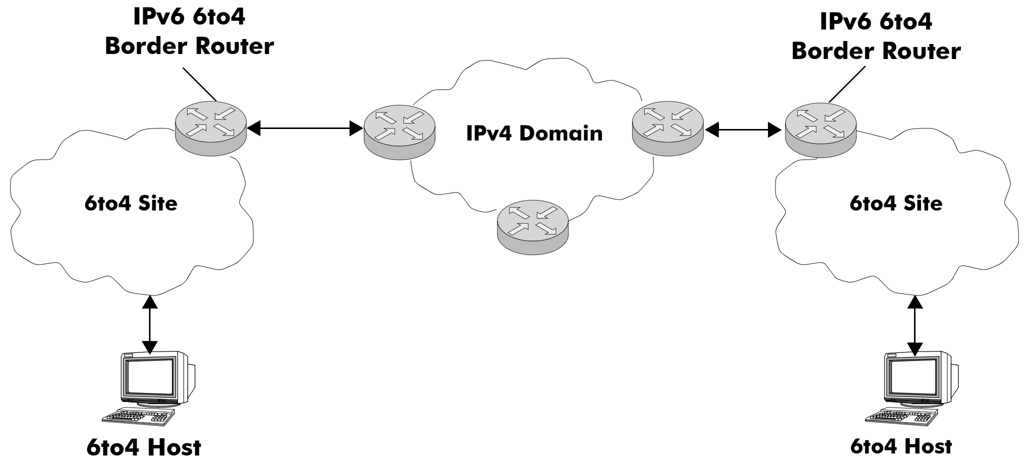
An IPv6 6to4 tunnel interface is identified by its assigned address, which is derived by combining a 6to4 well-known prefix (2002) with a globally unique IPv4 address and embedded as the first 48 bits of an IPv6 address. For example, 2002:d467:8a89::137/64, where D467:8A89 is the hex equivalent of the IPv4 address 212.103.138.137.

6to4 tunnel interfaces are configured on routers and identify a 6to4 site. Because 6to4 tunnels are point-to-multi-point in nature, any one 6to4 router can communicate with one or more other 6to4 routers across the IPv4 cloud. Two common scenarios for using 6to4 tunnels are described below.

6to4 Site to 6to4 Site over IPv4 Domain

In this scenario, isolated IPv6 sites have connectivity over an IPv4 network through 6to4 border routers. An IPv6 6to4 tunnel interface is configured on each border router and assigned an IPv6 address with the 6to4 well-known prefix, as described above. IPv6 hosts serviced by the 6to4 border router have at least one IPv6 router interface configured with a 6to4 address. Note that additional IPv6 interfaces or external IPv6 routing protocols are not required on the 6to4 border router.

The following diagram illustrates the basic traffic flow between IPv6 hosts communicating over an IPv4 domain:



In the above diagram:

1 6to4 hosts receive 6to4 prefix from Router Advertisement.

2 6to4 host sends IPv6 packets to 6to4 border router.

3 6to4 border router encapsulates IPv6 packets with IPv4 headers and sends to destination 6to4 border router over the IPv4 domain.

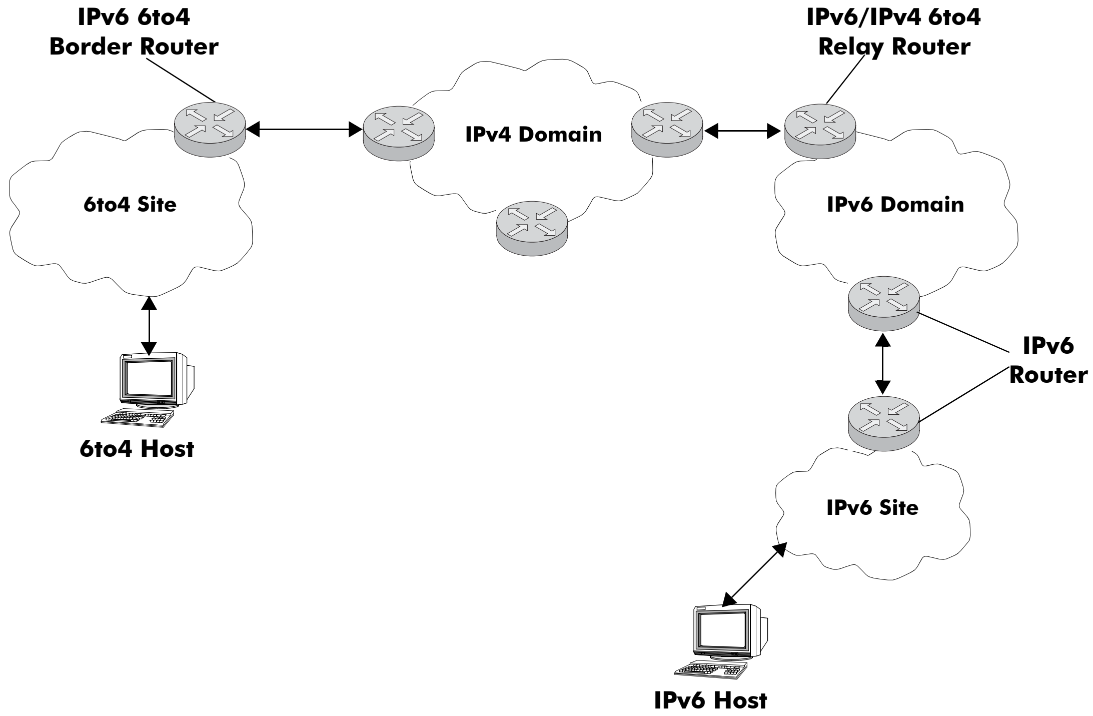
4 Destination 6to4 border router strips IPv4 header and forwards to 6to4 destination host.

6to4 Site to IPv6 Site over IPv4/IPv6 Domains

In this scenario, 6to4 sites have connectivity to native IPv6 domains through a relay router, which is connected to both the IPv4 and IPv6 domains. 6to4 border routers are still used by 6to4 sites for encapsulating/decapsulating host traffic and providing connectivity across the IPv4 domain. In addition, each border router has a default IPv6 route pointing to the relay router.

In essence, a relay router is a 6to4 border router in that a 6to4 tunnel interface is configured on the router. However, a native IPv6 router interface is also required on the relay router to transmit 6to4 traffic to/from IPv6 hosts connected to an IPv6 domain. Therefore, the relay router participates in both the IPv4 and IPv6 routing domains.

The following diagram illustrates the basic traffic flow between native IPv6 hosts and 6to4 sites:



In the above diagram:

1 6to4 relay router advertises a route to 2002::/16 on its IPv6 router interface.

2 IPv6 host traffic received by the relay router that has a next hop address that matches 2002::/16 is routed to the 6to4 tunnel interface configured on the relay router.

3 Traffic routed to the 6to4 tunnel interface is then encapsulated into IPv4 headers and sent to the destination 6to4 router over the IPv4 domain.

4 Destination 6to4 router strips IPv4 header and forwards to IPv6 destination host.

For more information about configuring an IPv6 6to4 tunnel interface, see "Configuring an IPv6 Interface" on page 14-10 and "Configuring IPv6 Tunnel Interfaces" on page 14-14. For more detailed information and scenarios using 6to4 tunnels, refer to RFC 3056.

Configured Tunnels

A configured tunnel is where the endpoint addresses are manually configured to create a point-to-point tunnel. This type of tunnel is similar to the 6to4 tunnel in that IPv6 packets are encapsulated in IPv4 headers to facilitate communication over an IPv4 network. The difference between the two types of tunnels is that configured tunnel endpoints require manual configuration, whereas 6to4 tunneling relies on an embedded IPv4 destination address to identify tunnel endpoints.

For more information about IPv6 configured tunnels, see "Configuring IPv6 Tunnel Interfaces" on page 14-14. For more detailed information about configured tunnels, refer to RFC 2893. Note that RFC 2893 also discusses automatic tunnels, which are not supported with this implementation of IPv6.

Configuring an IPv6 Interface

The **ipv6 interface** command is used to create an IPv6 interface for a VLAN or a tunnel. Note the following when configuring an IPv6 interface:

- A unique interface name is required for both a VLAN and tunnel interface.
- If creating a VLAN interface, the VLAN must already exist. See Chapter 5, "Configuring VLANs," for more information.
- If creating a tunnel interface, a tunnel ID or **6to4** is specified. Only one 6to4 tunnel is allowed per switch, so it is not necessary to specify an ID when creating this type of tunnel.
- If a tunnel ID is specified, then a configured tunnel interface is created. This type of tunnel requires additional configuration using the **ipv6 interface tunnel source destination** command. See "Configuring IPv6 Tunnel Interfaces" on page 14-14 for more information.
- The following configurable interface parameters are set to their default values, unless otherwise specified when the **ip interface** command is used:

IPv6 interface parameters	
mtu	**ra-reachable-time**
ra-send	**ra-retrans-timer**
ra-max-interval	**ra-default-lifetime**
ra-managed-config-flag	**ra-send-mtu**
ra-other-config-flag	

  Refer to the **ipv6 interface** command page in the *OmniSwitch CLI Reference Guide* for more details regarding these parameters.

- Each VLAN can have one IPv6 interface. Configuring both an IPv4 and IPv6 interface on the same VLAN is allowed. Note that VLAN interfaces of both types are not active until at least one port associated with the VLAN goes active.
- A link-local address is automatically configured for an IPv6 interface, except for 6to4 tunnels, when the interface is configured. For more information regarding how this address is formed, see "Autoconfiguration of IPv6 Addresses" on page 14-6.
- Assigning more than one IPv6 address to a single IPv6 interface is allowed.
- Assigning the same link-local address to multiple interfaces is allowed. Each global unicast prefix, however, can only exist on one interface. For example, if an interface for a VLAN 100 is configured with address 4100:1000::1/64, an interface for VLAN 200 cannot have address 4100:1000::2/64.
- Each IPv6 interface anycast address must also have a unique prefix. However, multiple devices may share the same anycast address prefix to identify themselves as members of the anycast group.

To create an IPv6 interface for a VLAN or configured tunnel, enter **ipv6 interface** followed by an interface name then **vlan** (or **tunnel**) followed by a VLAN ID (or tunnel ID). For example, the following two commands create an IPv6 interface for VLAN 200 and an interface for tunnel 35:

```
-> ipv6 interface v6if-v200 vlan 200
-> ipv6 interface v6if-tunnel-35 tunnel 35
```

To create an IPv6 interface for a 6to4 tunnel, the following command is used:

```
-> ipv6 interface v6if-6to4 tunnel 6to4
```

Use the **show ipv6 interface** command to verify the interface configuration for the switch. For more information about this command, see the *OmniSwitch CLI Reference Guide.*

Modifying an IPv6 Interface

The **ipv6 interface** command is also used to modify existing IPv6 interface parameter values. It is not necessary to first remove the interface and then create it again with the new values. The changes specified will overwrite existing parameter values. For example, the following command changes the router advertisement (RA) reachable time and the RA retransmit timer values for interface *v6if-v200:*

```
-> ipv6 interface v6if-v200 ra-reachable-time 60000 ra-retrans-time 2000
```

When an existing interface name is specified with the **ipv6 interface** command, the command modifies specified parameters for that interface. If an unknown interface name is entered along with an existing VLAN or tunnel parameter, a new interface is created with the name specified.

Removing an IPv6 Interface

To remove an IPv6 interface from the switch configuration, use the **no** form of the **ipv6 interface** command. Note that it is only necessary to specify the name of the interface, as shown in the following example:

```
-> no ipv6 interface v6if-v200
```

Assigning IPv6 Addresses

As was previously mentioned, when an IPv6 interface is created for a VLAN or a configured tunnel, an IPv6 link-local address is automatically created for that interface. This is also true when a device, such as a workstation, is connected to the switch.

Link-local addresses, although private and non-routable, enable interfaces and workstations to communicate with other interfaces and workstations that are connected to the same link. This simplifies getting devices up and running on the local network. If this level of communication is sufficient, assigning additional addresses is not required.

If it is necessary to identify an interface or device to the entire network, or as a member of a particular group, or enable an interface to perform routing functions, then configuring additional addresses (e.g., global unicast or anycast) is required.

Use the **ipv6 address** command to manually assign addresses to an existing interface (VLAN or tunnel) or device. For example, the following command assigns a global unicast address to the VLAN interface *v6if-v200*:

```
-> ipv6 address 4100:1000::20/64 v6if-v200
```

In the above example, 4100:1000:: is specified as the subnet prefix and 20 is the interface identifier. Note that the IPv6 address is expressed using CIDR notation to specify the prefix length. In the above example, **/64** indicates a subnet prefix length of 64 bits.

To use the MAC address of an interface or device as the interface ID, specify the **eui-64** option with this command. For example:

```
-> ipv6 address 4100:1000::/64 eui-64 v6if-v200
```

The above command example creates address 4100:1000::2d0:95ff:fe12:fab2/64 for interface *v6if-v200*.

Note the following when configuring IPv6 addresses:

- It is possible to assign more than one address to a single interface.
- Any field of an address may contain all zeros or all ones. The exception to this is that the interface identifier portion of the address cannot end in zero. If the **eui-64** option is specified with the **ipv6 address** command, this is not an issue.
- The EUI-64 interface identifier takes up the last 64-bits of the 128-bit IPv6 address. If the subnet prefix combined with the EUI-64 interface ID is longer than 128-bits, an error will occur and the address is not created.
- A subnet router anycast address is automatically created when a global unicast address is assigned to an interface. The anycast address is derived from the global address by adding an interface ID of all zeros to the prefix of the global address. For example, global address 4100:1000::20/64 generates the anycast address 4100:1000::/64.
- Devices, such as a PC, are eligible for stateless autoconfiguration of unicast addresses in addition to the link-local address. If this type of configuration is in use on the network, manual configuration of addresses is not required.
- IPv6 VLAN or tunnel interfaces are only eligible for stateless autoconfiguration of their link-local addresses. Manual configuration of addresses is required for all additional addresses.

See "IPv6 Addressing" on page 14-5 for an overview of IPv6 address notation. Refer to RFC 3513 for more technical address information.

Removing an IPv6 Address

To remove an IPv6 address from an interface, use the **no** form of the **ipv6 address** command.

```
-> no ipv6 address 4100:1000::20/64 v6if-v200
```

Note that the subnet router anycast address is automatically deleted when the last unicast address of the same subnet is removed from the interface.

Configuring IPv6 Tunnel Interfaces

There are two types of tunnels supported: 6to4 and configured. Both types facilitate the interaction of IPv6 with IPv4 networks by providing a mechanism for carrying IPv6 traffic over an IPv4 network infrastructure. This is an important function since it is more than likely that both protocols will need to coexist within the same network for some time.

A 6to4 tunnel is configured by creating an IPv6 6to4 tunnel interface on a router. This interface is then assigned an IPv6 address with an embedded well-known 6to4 prefix (e.g., 2002) combined with an IPv4 destination address. This is all done using the **ipv6 interface** and **ipv6 address** commands. For example, the following commands create a 6to4 tunnel interface:

```
-> ipv6 interface v6if-6to4-192 tunnel 6to4
-> ipv6 address 2002:d467:8a89::/48 v6if-6to4-192
```

In the above example, 2002 is the well-known prefix that identifies a 6to4 tunnel. The D467:8A89 part of the address that follows 2002 is the hex equivalent of the IPv4 address 212.103.138.137. Note that an IPv4 interface configured with the embedded IPv4 address is required on the switch. In addition, do not configure a private (e.g., 192.168.10.1), broadcast, or unspecified address as the embedded IPv4 address.

One of the main benefits of 6to4 tunneling is that no other configuration is required to identify tunnel endpoints. The router that the 6to4 tunnel interface is configured on, will encapsulate IPv6 packets in IPv4 headers and send them to the IPv4 destination address where they will be processed. This is particularly useful in situations where the IPv6 host is isolated.

The second type of tunnel supported is referred to as a configured tunnel. With this type of tunnel it is necessary to specify an IPv4 address for the source and destination tunnel endpoints. Note that if bidirectional communication is desired, then it is also necessary to create the tunnel interface at each end of the tunnel.

Creating an IPv6 configured tunnel involves the following general steps:

- Create an IPv6 tunnel interface using the **ipv6 interface** command.
- Associate an IPv4 source and destination address with the tunnel interface using the **ipv6 interface tunnel source destination** command. These addresses identify the tunnel endpoints.
- Associate an IPv6 address with the tunnel interface using the **ipv6 address** command.
- Configure a tunnel interface and associated addresses at the other end of tunnel.

The following example commands create the *v6if-tunnel-137* configured tunnel:

```
-> ipv6 interface v6if-tunnel-137 tunnel 1
-> ipv6 interface v6if-tunnel-137 tunnel source 212.103.138.137 destination
212.109.138.195
-> ipv6 address 4132:4000::/64 eui-64 v6if-tunnel-137
```

Note that RIPng is not supported over 6to4 tunnels but is allowed over configured tunnels. To use this protocol on a configured tunnel, a RIPng interface is created for the tunnel interface. For example, the following command creates an RIPng interface for tunnel v6if-tunnel-137:

```
-> ipv6 rip interface v6if-tunnel-137
```

Verifying the IPv6 Configuration

A summary of the show commands used for verifying the IPv6 configuration is given here:

show ipv6 interface	Displays the status and configuration of IPv6 interfaces.
show ipv6 tunnel	Displays IPv6 configured tunnel information and whether or not the 6to4 tunnel is enabled.
show ipv6 routes	Displays the IPv6 Forwarding Table.
show ipv6 prefixes	Displays IPv6 subnet prefixes used in router advertisements.
show ipv6 hosts	Displays the IPv6 Local Host Table.
show ipv6 neighbors	Displays the IPv6 Neighbor Table.
show ipv6 traffic	Displays statistics for IPv6 traffic.
show ipv6 icmp statistics	Displays ICMP6 statistics.
show ipv6 pmtu table	Displays the IPv6 Path MTU Table.
show ipv6 tcp ports	Displays TCP Over IPv6 Connection Table. Contains information about existing TCP connections between IPv6 endpoints.
show ipv6 udp ports	Displays the UDP Over IPv6 Listener Table. Contains information about UDP/IPv6 endpoints

For more information about the displays that result from these commands, see the *OmniSwitch CLI Reference Guide*.

15 Configuring RIP

Routing Information Protocol (RIP) is a widely used Interior Gateway Protocol (IGP) that uses hop count as its routing metric. RIP-enabled routers update neighboring routers by transmitting a copy of their own routing table. The RIP routing table uses the most efficient route to a destination, that is, the route with the fewest hops and longest matching prefix.

The switch supports RIP version 1 (RIPv1), RIP version 2 (RIPv2), and RIPv2 that is compatible with RIPv1. It also supports text key and MD5 authentication, on an interface basis, for RIPv2.

In This Chapter

This chapter describes RIP and how to configure it through the Command Line Interface (CLI). It includes instructions for configuring basic RIP routing, and fine-tuning RIP using optional RIP configuration parameters (e.g., RIP send/receive option, RIP interface metric). It also details RIP redistribution, which allows a RIP network to exchange routing information with networks running different protocols (e.g., OSPF, BGP). CLI commands are used in the configuration examples; for more details about the syntax of commands, see the *OmniSwitch CLI Reference Guide*.

This chapter provides an overview of RIP and includes information about the following procedures:

- RIP Routing
  - Loading RIP (see page 15-6)
  - Enabling RIP (see page 15-6)
  - Creating a RIP Interface (see page 15-7)
  - Enabling a RIP Interface (see page 15-7)
- RIP Options
  - Configuring the RIP Forced Hold-down Interval (see page 15-9)
  - Enabling a RIP Host Route (see page 15-9)
- RIP Redistribution
  - Enabling RIP Redistribution (see page 15-10)
  - Configuring RIP Redistribution Policies (see page 15-10)
  - Configuring RIP Redistribution Filters (see page 15-11)
- RIP Security
  - Configuring Authentication Type (see page 15-14)
  - Configuring Passwords (see page 15-15)

RIP Specifications

RFCs Supported	RFC 1058–RIP v1 RFC 2453–RIP v2 RFC 1722–RIP v2 Protocol Applicability Statement RFC 1724–RIP v2 MIB Extension
Maximum Number of RIP Routes	10240

RIP Defaults

The following table lists the defaults for RIP configuration through the **ip rip** command.

Description	Command	Default
RIP Status	**ip rip status**	disable
RIP Forced Hold-down Interval	**ip rip force-holddowntimer**	0
RIP Interface Metric	**ip rip interface metric**	1
RIP Interface Send Version	**ip rip interface send-version**	v2
RIP Interface Receive Version	**ip rip interface recv-version**	both
RIP Host Route	**ip rip host-route**	enable
RIP Route Tag	**ip rip route-tag**	0
Redistribution Status	**ip rip redist status**	disable
Redistribution Metric	**ip rip redist metric**	0
Redistribution Filter Effect	**ip rip redist-filter effect**	permit
Redistribution Filter Metric	**ip rip redist-filter metric**	0
Redistribution Filter Control	**ip rip redist-filter redist-control**	all-subnets
Redistribution Filter Route Tag	**ip rip redist-filter route-tag**	0
RIP Interface Authentication	**ip rip interface auth-type**	none

Quick Steps for Configuring RIP Routing

To forward packets to a device on a different VLAN, you must create a router port on each VLAN. To route packets using RIP, you must enable RIP and create a RIP interface on the router port. The following steps show you how to enable RIP routing between VLANs "from scratch". If active VLANs and router ports have already been created on the switch, go to Step 7.

1 Create VLAN 1 with a description (e.g., VLAN 1) using the **vlan** command. For example:

```
-> vlan 1 name "VLAN 1"
```

2 Create VLAN 2 with a description (e.g., VLAN 2) using the **vlan** command. For example:

```
-> vlan 2 name "VLAN 2"
```

3 Assign an active port to VLAN 1 using the **vlan port default** command. For example, the following command assigns port 1 on slot 1 to VLAN 1:

```
-> vlan 1 port default 1/1
```

4 Assign an active port to VLAN 2 using the **vlan port default** command. For example, the following command assigns port 2 on slot 1 to VLAN 2:

```
-> vlan 2 port default 1/2
```

5 Configure an IP interface to enable IP routing on a VLAN by using **ip interface**. For example:

```
-> ip interface vlan-1 address 171.10.1.1 vlan 1
```

6 Configure an IP interface to enable IP routing on a VLAN by using **ip interface**. For example:

```
-> ip interface vlan-2 address 171.11.1.1 vlan 2
```

7 Load RIP into switch memory using the **ip load rip** command. For example:

```
-> ip load rip
```

8 Enable RIP on the switch using the **ip rip status** command. For example:

```
-> ip rip status enable
```

9 Create a RIP interface on VLAN 1 using the **ip rip interface** command. For example:

```
-> ip rip interface 171.10.1.1
```

10 Enable the RIP interface using the **ip rip interface status** command. For example:

```
-> ip rip interface 171.10.1.1 status enable
```

11 Create a RIP interface on VLAN 2 using the **ip rip interface** command. For example:

```
-> ip rip interface 171.11.1.1
```

12 Enable the RIP interface using the **ip rip interface status** command. For example:

```
-> ip rip interface 171.11.1.1 status enable
```

13 Enable redistribution of local routes on the switch using the **ip rip redist** command. For example:

```
-> ip rip redist local
```

14 Use the **ip rip redist-filter** command to redistribute all local routes. For example:

```
-> ip rip redist-filter local 0.0.0.0 0.0.0.0
```

15 Enable RIP redistribution using the **ip rip redist status** command. For example:

```
-> ip rip redist status enable
```

Note. For more information on VLANs and router ports, see Chapter 5, “Configuring VLANs.”

RIP Overview

In switching, traffic may be transmitted from one media type to another within the same VLAN. Switching happens at Layer 2, the link layer; routing happens at Layer 3, the network layer. In IP routing, traffic can be transmitted across VLANs. When IP routing is enabled, the switch uses routing protocols to build routing tables that keep track of stations in the network and decide the best path for forwarding data. When the switch receives a packet to be routed, it strips off the MAC header and examines the IP header. It looks up the source/destination address in the routing table, and then adds the appropriate MAC address to the packet. Calculating routing tables and stripping/adding MAC headers to packets is performed by switch software.

IP is associated with several Layer 3 routing protocols. RIP is built into the base code loaded onto the switch. Others are part of Alcatel’s Advanced Routing Software. IP supports the following IP routing protocols:

- RIP—An IGP that defines how routers exchange information. RIP makes routing decisions using a “least-cost path” method. RIPv1 and RIPv2 services allow the switch to learn routing information from neighboring RIP routers. For more information and instructions for configuring RIP, see “RIP Routing” on page 15-5.

- Open Shortest Path First (OSPF)—An IGP that provides a routing function similar to RIP but uses different techniques to determine the best route for a datagram. OSPF is part of Alcatel’s Advanced Routing Software. For more information see the “Configuring OSPF” chapter in the *OmniSwitch 7700/7800/8800 Advanced Routing Configuration Guide*.

When RIP is initially enabled on a switch, it issues a request for routing information, and listens for responses to the request. If a switch configured to supply RIP hears the request, it responds with a response packet based on information in its routing database. The response packet contains destination network addresses and the routing metric for each destination. When a RIP response packet is received, RIP takes the information and rebuilds the switch’s routing database, adding new routes and “better” (lower metric) routes to destinations already listed in the database.

RIP uses a hop count metric to measure the distance to a destination. In the RIP metric, a switch advertises directly connected networks at a metric of 1. Networks that are reachable through one other gateway are 2 hops, networks that are reachable through two gateways are 3 hops, etc. Thus, the number of hops (or hop count) along a path from a given source to a given destination refers to the number of networks that are traversed by a datagram along that path. When a switch receives a routing update that contains a new or changed destination network entry, the switch adds one to the metric value indicated in the update and enters the network in the routing table. After updating its routing table, the switch immediately begins transmitting routing updates to inform other network switches of the change. These updates are sent independently of the regularly scheduled updates. By default, RIP packets are broadcast every 30 seconds, even if no change has occurred anywhere in a route or service.

RIP deletes routes from the database if the next switch to that destination says the route contains more than 15 hops. In addition, all routes through a gateway are deleted by RIP if no updates are received from that gateway for a specified time period. If a gateway is not heard from for 180 seconds, all routes from that gateway are placed in a hold-down state. If the hold-down timer value is exceeded, the routes are deleted from the routing database. These intervals also apply to deletion of specific routes.

RIP Version 2

RIP version 2 (RIPv2) adds additional capabilities to RIP. Not all RIPv2 enhancements are compatible with RIPv1. To avoid supplying information to RIPv1 routes that could be misinterpreted, RIPv2 can only use non-compatible features when its packets are multicast. Multicast is not supported by RIPv1. On interfaces that are not compatible with IP multicast, the RIPv1-compatible packets used do not contain potentially confusing information. RIPv2 enhancements are listed below.

- Next Hop—RIPv2 can advertise a next hop other than the switch supplying the routing update. This capability is useful when advertising a static route to a silent switch not using RIP, since packets passing through the silent switch do not have to cross the network twice.

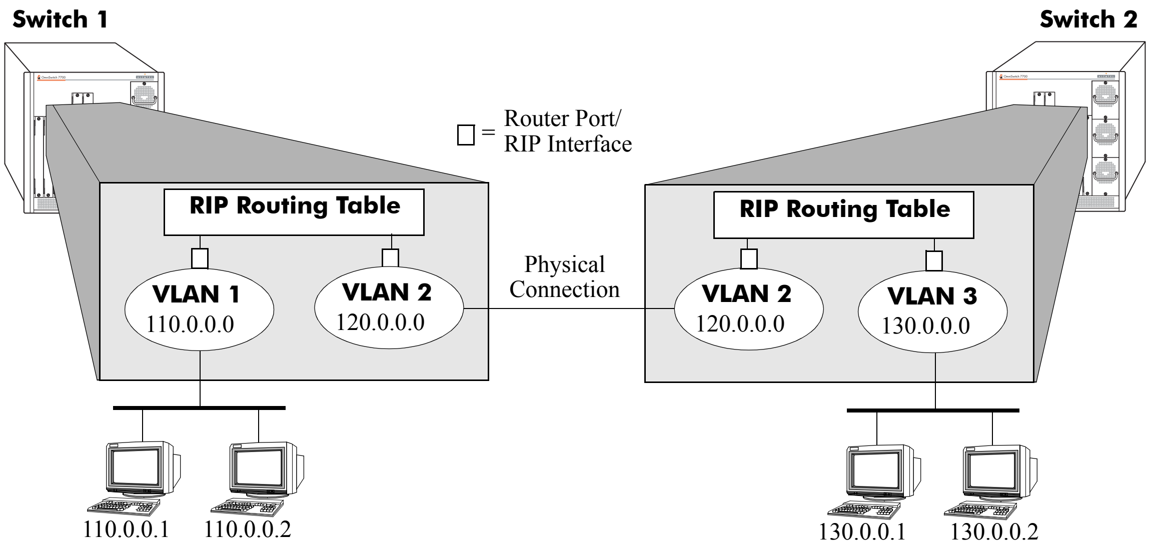
- Network Mask—RIPv1 assumes that all subnetworks of a given network have the same network mask. It uses this assumption to calculate the network masks for all routes received. This assumption prevents subnets with different netmasks from being included in RIP packets. RIPv2 adds the ability to specify the network mask with each network in a packet. Because RIPv1 switches ignore the network mask in RIPv2 packets, their calculation of the network mask could possibly be wrong. For this reason, RIPv1-compatible RIPv2 packets cannot contain networks that would be misinterpreted by RIPv1. These networks must only be provided in native RIPv2 packets that are multicast.

- Authentication—RIPv2 packets can contain an authentication key that may be used to verify the validity of the supplied routing data. Authentication may be used in RIPv1-compatible RIPv2 packets, but RIPv1 switches will ignore authentication information. The authentication is a simple password in which an authentication key of up to 16 characters is included in the packet. If this key does not match the configured authentication key, the packet is discarded. For more information on RIP authentication, see “RIP Security” on page 15-14.

- IP Multicast—IP Multicast Switching (IPMS) is a one-to-many communication technique employed by emerging applications such as video distribution, news feeds, netcasting, and resource discovery. Unlike unicast, which sends one packet per destination, multicast sends one packet to all devices in any subnetwork that has at least one device requesting the multicast traffic. For more information on IPMS, see Chapter 26, “Configuring IP Multicast Switching.”

RIP Routing

IP routing requires IP router ports to be configured on VLANs and a routing protocol to be enabled and configured on the switch. RIP also requires a RIP interface to be created and enabled on the routing port. In the illustration below, a router port and RIP interface have been configured on each VLAN. Therefore, workstations connected to ports on VLAN 1 on Switch 1 can communicate with VLAN 2; and workstations connected to ports on VLAN 3 on Switch 2 can communicate with VLAN 2. Also, ports from both switches have been assigned to VLAN 2, and a physical connection has been made between the switches. Therefore, workstations connected to VLAN 1 on Switch 1 can communicate with workstations connected to VLAN 3 on Switch 2.



RIP Routing

Loading RIP

When the switch is initially configured, RIP must be loaded into switch memory. Use the **ip load rip** command to load RIP.

To remove RIP from switch memory, you must manually edit the **boot.cfg** file. The **boot.cfg** file is an ASCII text-based file that controls many of the switch parameters. Open the file and delete all references to RIP. You must reboot the switch when this is complete.

Note. In simple networks where only IP forwarding is required, you may not want to use RIP. If you are not using RIP, it is best not to load it to save switch resources.

Enabling RIP

RIP is disabled by default. Use the **ip rip status** command to enable RIP routing on the switch. For example:

```
-> ip rip status enable
```

Use the **ip rip status disable** command to disable RIP routing on the switch. Use the **show ip rip** command to display the current RIP status.

Creating a RIP Interface

You must create a RIP interface on a VLAN's IP router port to enable RIP routing. Enter the **ip rip interface** command followed by the IP address of the VLAN router port. For example, to create a RIP interface on a router port with an IP address of 171.15.0.1 you would enter:

```
-> ip rip interface 171.15.0.1
```

Use the **no ip rip interface** command to delete a RIP interface. Use the **show ip rip interface** command to display configuration and error information for a RIP interface.

Note. You can create a RIP interface even if an IP router port has not been configured. However, RIP will not function unless a RIP interface is created and enabled on an IP router port. For more information on VLANs and router ports, see "Configuring VLANs" on page 5-1.

Enabling a RIP Interface

Once you have created a RIP interface, you must enable it to enable RIP routing. Use the **ip rip interface status** command followed by the interface IP address to enable a RIP interface. For example, to enable RIP routing on RIP interface 171.15.0.1 you would enter:

```
-> ip rip interface 171.15.0.1 status enable
```

To disable a RIP interface, use the **disable** keyword with the **ip rip interface status** command. For example to disable RIP routing on RIP interface 171.15.0.1 you would enter:

```
-> ip rip interface 171.15.0.1 status disable
```

Configuring the RIP Interface Send Option

The RIP send option defines the type(s) of RIP packets that the interface will send. Using this command will override RIP default behavior. Other devices must be able to interpret the information provided by this command or routing information will not be properly exchanged between the switch and other devices on the network.

Use the **ip rip interface send-version** command to configure an individual RIP interface send option. Enter the IP address of the RIP interface, then enter a send option. For example, to configure RIP interface 172.22.2.115 to send only RIPv1 packets you would enter:

```
-> ip rip interface 172.22.2.115 send-version v1
```

The send options are:

- **v1.** Only RIPv1 packets will be sent by the switch.
- **v2.** Only RIPv2 packets will be sent by the switch.
- **v1compatible.** Only RIPv2 broadcast packets (not multicast) will be sent by the switch.
- **none.** Interface will not forward RIP packets.

The default RIP send option is **v2**.

Use the **show ip rip interface** command to display the current interface send option.

Configuring the RIP Interface Receive Option

The RIP receive option defines the type(s) of RIP packets that the interface will accept. Using this command will override RIP default behavior. Other devices must be able to interpret the information provided by this command or routing information will not be properly exchanged between the switch and other devices on the network.

Use the **ip rip interface recv-version** command to configure an individual RIP interface receive option. Enter the IP address of the RIP interface, then enter a receive option. For example, to configure RIP interface 172.22.2.115 to receive only RIPv1 packets you would enter:

```
-> ip rip interface 172.22.2.115 recv-version v1
```

The receive options are:

- **v1.** Only RIPv1 packets will be received by the switch.
- **v2.** Only RIPv2 packets will be received by the switch.
- **both.** Both RIPv1 and RIPv2 packets will be received by the switch.
- **none.** Interface ignores any RIP packets received.

The default RIP receive option is **both**.

Configuring the RIP Interface Metric

You can set priorities for routes generated by a switch by assigning a metric value to routes generated by that switch's RIP interface. For example, routes generated by a neighboring switch may have a hop count of 1. However, you can lower the priority of routes generated by that switch by increasing the metric value for routes generated by the RIP interface.

Note. When you configure a metric for a RIP interface, this metric cost is added to the metric of the incoming route.

Use the **ip rip interface metric** command to configure the RIP metric or cost for routes generated by a RIP interface. Enter the IP address of the RIP interface as well as a metric value. For example, to set a metric value of 2 for RIP interface 171.15.0.1 you would enter:

```
-> ip rip interface 171.15.0.1 metric 2
```

The valid metric range is **1** to **15**. The default is **1**.

Use the **show ip rip interface** command to display the current interface metric.

Configuring the RIP Interface Route Tag

Use the **ip rip route-tag** command to configure a route tag value for routes generated by the RIP interface. This value is used to set priorities for RIP routing. Enter the command and the route tag value. For example, to set a route tag value of 1 you would enter:

```
-> ip rip route-tag 1
```

The valid route tag value range is **1** to **2147483647**. The default is **0**.

Use the **show ip rip** command to display the current route tag value.

RIP Options

The following sections detail procedures for configuring RIP options. RIP must be loaded and enabled on the switch before you can configure any of the RIP configuration options.

Configuring the RIP Forced Holddown Interval

The RIP forced holddown timer value defines an amount of time, in seconds, during which routing information regarding better paths is suppressed. A route enters into a forced holddown state when an update packet is received that indicates the route is unreachable and when this timer is set to a non-zero value. After this timer has expired and if the value is less that 120 seconds, the route enters a holddown state for the rest of the period until the remainder of the 120 seconds has also expired. During this time the switch will accept any advertisements for better paths that are received.

Note that the forced holddown timer is *not* the same as the RIP holddown timer. The RIP holddown timer is fixed at 120 seconds and is not configurable. The forced holddown timer defines a separate interval that overlaps the holddown state. During the forced holddown timer interval, the switch will not accept *better* routes from other gateways.

Use the **ip rip force-holddowntimer** command to configure the interval during which a RIP route remains in a forced hold-down state. Enter the command and the forced hold-down interval value, in seconds. For example, to set a forced hold-down interval value of 10 seconds you would enter:

```
-> ip rip force-holddowntimer 10
```

The valid forced hold-down timer range is **0** to **120**. The default is **0**.

Use the **show ip rip** command to display the current forced hold-down timer value.

Enabling a RIP Host Route

A host route differs from a network route, which is a route to a specific network. This command allows a direct connection to the host without using the RIP table. If a switch is directly attached to a host on a network, use the **ip rip host-route** command to enable a default route to the host. For example:

```
-> ip rip host-route
```

The default is to enable a default host route.

Use the **no ip rip host-route** command to disable the host route. Use the **show ip rip** command to display the current host route status.

RIP Redistribution

Redistribution provides a way to exchange routing information between RIP networks and OSPF and BGP networks; and also redistributes local and static routes into RIP. Basically, redistribution makes a non-RIP route look like a RIP route. Configuring RIP redistribution consists of the following tasks:

1. Enabling RIP Redistribution
2. Configuring a RIP Redistribution Policy

3 Configuring a RIP Redistribution Filter

- Creating a Filter
- Configuring a Redistribution Filter Action (optional)
- Configuring a Redistribution Metric (optional).

Enabling RIP Redistribution

Use the **ip rip redist status** command to enable/disable redistribution. For example, to enable RIP redistribution you would enter:

```
-> ip rip redist status enable
```

RIP redistribution is disabled by default.

Use the **ip rip redist status disable** command to disable redistribution. Use the **show ip rip** command to display RIP redistribution status.

Configuring a RIP Redistribution Policy

After enabling RIP redistribution, configure a policy that defines the route types that will be redistributed into RIP. Only the route types you configure will be redistributed into RIP. When you configure a redistribution policy, RIP is automatically enabled.

Use the **ip rip redist** command to define the route types that will be redistributed. Enter the command, then enter the route type. For example, to redistribute OSPF routes into RIP you would enter:

```
-> ip rip redist ospf
```

The redistribution route types are:

- **local.** Redistribute local routes into RIP.
- **static.** Redistribute static routes into RIP.
- **ospf.** Redistribute routes learned through OSPF into RIP.
- **bgp.** Redistribute routes learned through BGP into RIP.

Use the **no ip rip redist** command to delete a redistribution policy. For example, to “turn off” redistribution of OSPF routes you would enter:

```
-> no ip rip redist ospf
```

Note. If you are configuring more than one route type, you must repeat the command for each one.

Use the **show ip rip redist** command to display the status of RIP policies.

Configuring a Redistribution Metric

When redistributing routes into RIP, the metric for the redistributed route is calculated as a summation of the route's metric and the corresponding metric in the redistribution policy. This is the case when the matching filter metric is 0 (the default). However, if the matching redistribution filter metric is set to a non-zero value, the redistributed route's metric is set to the filter metric. This gives better control of the metric when redistributing non-RIP routes into RIP.

Note that if the metric calculated for the redistributed route, as described above, is *greater* than 15 (RIP_UNREACHABLE) or *greater* than the metric of an existing pure RIP route, the new route is not redistributed.

Use the **ip rip redist metric** command to configure the RIP metric or cost for a route type. Enter the command, specify the route type to be redistributed, then enter a metric value. For example:

```
-> ip rip redist ospf metric 2
```

The valid metric range is **0** to **15** (default is **0**).

Note. You must configure a redistribution policy before configuring a redistribution metric for that type. See "Configuring a RIP Redistribution Policy" on page 15-10 for information on configuring redistribution policies. If you are configuring a metric value for more than one route type, you must repeat the command for each one.

Configuring a RIP Redistribution Filter

After configuring a redistribution policy (e.g., OSPF), you must specify what routes will be redistributed by configuring a redistribution filter. Only routes matching the policy and destination specified in the filter will be redistributed into RIP. Creating a RIP redistribution filter consists of the following steps:

- Creating a Redistribution Filter
- Configuring the Redistribution Filter Action (optional)
- Configuring the Redistribution Filter Metric (optional)
- Configuring the Redistribution Filter Route Control Action (optional)
- Configuring a Redistribution Filter Route Tag (optional).

Note. You must first configure a redistribution policy before configuring a filter for a route type. See "Configuring a RIP Redistribution Policy" on page 15-10 for information on configuring redistribution policies.

Creating a Redistribution Filter

Use the **ip rip redist-filter** command to create a RIP redistribution filter. Enter the command, the route type, and destination IP address and mask of the traffic you want to redistribute. Only routes matching the policy and destination specified in the filter will be redistributed into RIP and passed to the destination. For example to redistribute OSPF routes destined for the 10.0.0.0 network you would enter:

```
-> ip rip redist-filter ospf 10.0.0.0 255.0.0.0
```

Note. A network/subnetwork of 0.0.0.0. 0.0.0.0 will redistribute all routes for the configured route type.

Use the **no ip rip redist-filter** command to delete a filter. For example, to “turn off” redistribution for OSPF routes to the 10.0.0.0 network you would enter:

```
-> no ip rip redist-filter ospf 10.0.0.0 255.0.0.0
```

Use the **show ip rip redist-filter** command to display the currently-configured redistribution filters.

Note. Local interfaces will not be added to the RIP routing table unless RIP redistribution is enabled and a filter is added for the local protocol.

Configuring a Redistribution Filter Action

By default, redistribution filters allow (permit) routes that match the criteria specified in the filter to be redistributed. However, you can use the redistribution filter action feature to “fine-tune” a filter. You may want to redistribute all routes to a network except routes destined for a particular subnet. In this case, you would “permit” all traffic to the network but “deny” traffic to a particular subnet.

Use the **ip rip redist-filter effect** command to configure the redistribution filter action. Enter the command, specify the route type to be redistributed, enter the destination IP address/mask, then enter whether to permit redistribution (**permit**) or deny redistribution (**deny**).

For example, if you wanted to redistribute all OSPF routes to the 172.22.0.0 network except routes to subnetwork 3 you would used the following commands:

```
-> ip rip redist-filter ospf 172.22.0.0 255.255.0.0 effect permit

-> ip rip redist-filter ospf 172.22.3.0 255.255.255.0 effect deny
```

Configuring a Redistribution Filter Metric

You can prioritize redistribution of route types to a network by assigning a metric value to a route type(s). The default redistribution filter metric is 1. However, you can lower the priority of a route type by increasing its metric value. For example, if you want to give priority to OSPF routes to a particular network, you would set the metric value for the other route types to 2.

Use the **ip rip redist-filter metric** command to configure a metric value. Enter the command, specify the route type to be redistributed, enter the destination IP address/mask, then enter the metric value. For example, if you wanted to lower the priority of OSPF routes to a network, and all other route types were set to the default metric of 1, you would to set a metric value of 2 for OSPF routes destined for that network.

```
-> ip rip redist-filter metric ospf 172.22.0.0 255.255.0.0 metric 2
```

Note. If you are configuring a metric value for more than one route type, you must repeat the command for each one.

The redistribution filter metric range is **0** to **15**. The default is **0**.

Configuring the Redistribution Filter Route Control Action

In certain cases, the specified route to be filtered will be either an aggregate route or a subnet. In these cases, the route may be comprised of several routes. It is possible to redistribute these routes separately or not using the **ip rip redist-filter redist-control** command. Enter the command, specify the route type to be redistributed, enter the destination IP address/mask, then enter a route control action:

- **all-subnets.** Redistributes all subnet routes that match this filter, if permitted (default).
- **aggregate.** Redistributes an aggregate route if there are one or more routes that match this filter.
- **no-subnets.** Redistributes only those routes that exactly match the redistribution filter.

For example, if the route being filtered is an aggregate or subnet route, and the routes that comprise the aggregate or subnet route should not be redistributed, enter the **ip rip redist-filter redist-control** command, and the **no-subnets** keyword.

```
-> ip rip redist-filter ospf 172.22.0.0 255.255.0.0 redist-control no-subnets
```

Note. By default, filters are set to allow subnet routes to be advertised. If this is the filter action desired, it is not necessary to use the **redist-control** keyword.

Configuring a Redistribution Filter Route Tag

The redistribution route tag specifies the route tag with which routes matching a filter are redistributed into RIP. The default value is zero (0), which means that the route tag used will be the one in the route, if specified.

Use the **ip rip redist-filter route-tag** command to configure a redistribution route tag. Enter the command, specify the route type to be redistributed, enter the destination IP address/mask, then enter the route tag value. For example, if you wanted to configure a route tag value of 1 for OSPF routes to the 172.22.0.0 network you would enter:

```
-> ip rip redist-filter ospf 172.22.0.0 255.255.0.0 route-tag 1
```

RIP Security

By default, there is no authentication used for a RIP. However, you can configure a password for a RIP interface. To configure a password, you must first select the authentication type (simple or MD5), then configure a password.

Configuring Authentication Type

If simple or MD5 authentication is used, both switches on either end of a link must share the same password. Use the **ip rip interface auth-type** command to configure the authentication type. Enter the IP address of the RIP interface, then enter an authentication type:

- **none.** No authentication will be used.
- **simple.** Simple authentication will be used.
- **md5**. MD5 authentication will be used.

For example, to configure RIP interface 172.22.2.115 for simple authentication you would enter:

```
-> ip rip interface 172.22.2.115 auth-type simple
```

To configure RIP interface 172.22.2.115 for MD5 authentication you would enter:

```
-> ip rip interface 172.22.2.115 md5 auth-type md5
```

Configuring Passwords

If you configure simple or MD5 authentication, you must configure a text string that will be used as the password for the RIP interface. If a password is used, all switches that are intended to communicate with each other must share the same password.

After configuring the interface for simple authentication as described above, configure the password for the interface using the **ip rip interface auth-key** command. Enter the IP address of the RIP interface, then enter a 16-byte text string. For example to configure a password “nms” you would enter:

```
-> ip rip interface 172.22.2.115 auth-key nms
```

Verifying the RIP Configuration

A summary of the show commands used for verifying the RIP configuration is given here:

show ip rip	Displays RIP status and general configuration parameters (e.g., forced hold-down timer).
show ip rip routes	Displays the RIP routing database. The routing database contains all of the routes learned through RIP.
show ip rip interface	Displays RIP interface status and configuration.
show ip rip peer	Displays active RIP neighbors (peers).
show ip rip redist	Displays general RIP redistribution parameters.
show ip rip redist-filter	Displays currently-configured RIP redistribution filters.

For more information about the displays that result from these commands, see the *OmniSwitch CLI Reference Guide*.

16 Configuring RDP

The Router Discovery Protocol (RDP) is an extension of ICMP that allows end hosts to discover routers on their networks. This implementation of RDP supports the router requirements as defined in RFC 1256.

In This Chapter

This chapter describes the RDP feature and how to configure RDP parameters through the Command Line Interface (CLI). CLI commands are used in the configuration examples; for more details about the syntax of commands, see the *OmniSwitch CLI Reference Guide*.

The following procedures are described:

- "Enabling/Disabling RDP" on page 16-8.
- "Creating an RDP Interface" on page 16-8.
- "Specifying an Advertisement Destination Address" on page 16-9.
- "Defining the Advertisement Interval" on page 16-9.
- "Setting the Advertisement Lifetime" on page 16-10.
- "Setting the Preference Levels for Router IP Addresses" on page 16-11.
- "Verifying the RDP Configuration" on page 16-11.

RDP Specifications

RFCs Supported	RFC 1256–ICMP Router Discovery Messages
Router advertisements	Supported
Host solicitations	Only responses to solicitations supported in this release.
Maximum number of RDP interfaces per switch	One for each available IP interface configured on the switch.
Advertisement destination addresses	224.0.0.1 (all systems multicast) 255.255.255.255 (broadcast)

RDP Defaults

Parameter Description	CLI Command	Default Value/Comments
RDP status for the switch	**ip router-discovery**	Disabled
RDP status for switch interfaces (router VLAN IP addresses)	**ip router-discovery interface**	Disabled
Advertisement destination address for an active RDP interface.	**ip router-discovery interface advertisement-address**	All systems multicast (224.0.0.1)
Maximum time between advertisements sent from an active RDP interface	**ip router-discovery interface max-advertisement-interval**	600 seconds
Minimum time between advertisements sent from an active RDP interface	**ip router-discovery interface min-advertisement-interval**	450 seconds (0.75 * maximum advertisement interval)
Maximum time IP addresses contained in an advertisement packet are considered valid	**ip router-discovery interface advertisement-lifetime**	1800 seconds (3 * maximum advertisement interval)
Preference level for IP addresses contained in an advertisement packet	**ip router-discovery interface preference-level**	0

Quick Steps for Configuring RDP

Configuring RDP involves enabling RDP operation on the switch and creating RDP interfaces to advertise VLAN router IP addresses on the LAN. There is no order of configuration involved. For example, it is possible to create RDP interfaces even if RDP is not enabled on the switch.

The following steps provide a quick tutorial on how to configure RDP. Each step describes a specific operation and provides the CLI command syntax for performing that operation.

1 Enable RDP operation on the switch.

```
-> ip router-discovery enable
```

Note. *Optional.* To verify the global RDP configuration for the switch, enter the **show ip router-discovery** command. The display is similar to the one shown below:

```
-> show ip router-discovery
Status                     = Enabled,
RDP uptime                 = 161636 secs
#Packets Tx                = 4,
#Packets Rx                = 0,
#Send Errors               = 0,
#Recv Errors               = 0,
```

For more information about this command, refer to the “RDP Commands” chapter in the *OmniSwitch CLI Reference Guide*.

2 Create an RDP interface for an IP router interface. In this example, an RDP interface is created for the IP router interface named Marketing (note that IP interfaces are referenced by their name):

```
-> ip router-discovery interface Marketing enable
```

3 When an RDP interface is created, default values are set for the interface advertisement destination address, transmission interval, lifetime, and preference level parameters. If you want to change the default values for these parameters, see “Creating an RDP Interface” on page 16-8 for more information.

Note. *Optional.* To verify the RDP configuration for all RDP interfaces, enter the **show ip router-discovery interface** command. The display is similar to the one shown below:

```
-> show ip router-discovery interface
                       IP i/f   RDP i/f  VRRP i/f      Next   #Pkts
         Name          status   status  status(#mast)  Advt sent recvd
---------------------+--------+--------+--------------+----+----------
Marketing             Disabled  Enabled Disabled(0)    9    0   0
Finance IP Network    Disabled  Enabled Disabled(0)    3    0   0
Accounting            Enabled   Enabled Enabled(2)     443  3   0
```

To verify the configuration for a specific RDP interface, specify the interface IP router interface name when using the **show ip router-discovery interface** command. The display is similar to the one shown below.

```
-> show ip router-discovery interface Marketing
Name                                    = Marketing,
IP Address                              = 11.255.4.1,
IP Mask                                 = 255.0.0.0,
IP Interface status                     = Enabled,
RDP Interface status                    = Enabled,
VRRP Interface status                   = Disabled,
Advertisement address                   = 224.0.0.1,
Max Advertisement interval              = 600 secs,
Min Advertisement interval              = 450 secs,
Advertisement lifetime                  = 1800 secs,
Preference Level                        = 0x0,
#Packets sent                           = 3,
#Packets received                       = 0
```

For more information about this command, refer to the “RDP Commands” chapter in the *OmniSwitch CLI Reference Guide*.

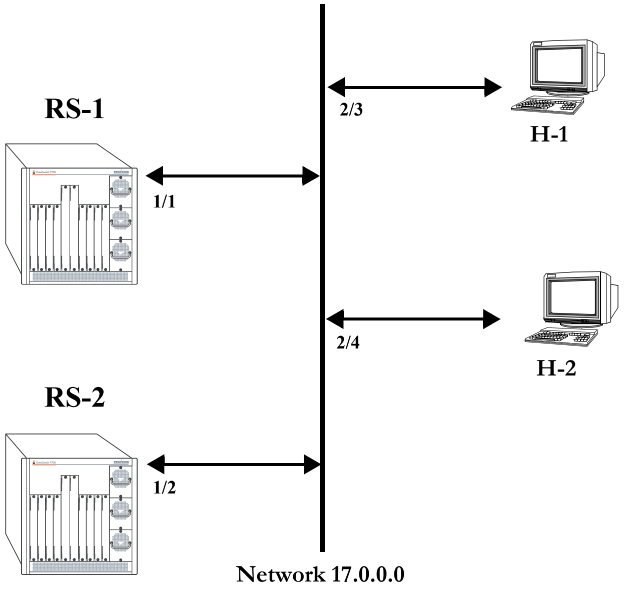
RDP Overview

End hosts (clients) sending traffic to other networks need to forward their traffic to a router. In order to do this, hosts need to find out if one or more routers exist on their LAN and learn their IP addresses. One way to discover neighboring routers is to manually configure a list of router IP addresses that the host reads at startup. Another method available involves listening to routing protocol traffic to gather a list of router IP addresses.

RDP provides an alternative method for hosts to discover routers on their network that involves the use of ICMP advertisement and solicitation messages. Using RDP, hosts attached to multicast or broadcast networks send solicitation messages when they start up. Routers respond to solicitation messages with an advertisement message that contains the router IP addresses. In addition, routers send advertisement messages when their RDP interface becomes active and then subsequently at random intervals.

When a host receives a router advertisement message, it adds the IP addresses contained in the message to its list of default router gateways in the order of preference. As a result, the list of router IP addresses is dynamically created and maintained, eliminating the need for manual configuration of such a list. In addition, hosts do not have to recognize many different routing protocols to discover router IP addresses.

The following diagram illustrates an example of using RDP in a typical network configuration:



When interfaces 2/3 and 2/4 on hosts H-1 and H-2, respectively, become active, they transmit router solicitation ICMP messages on Network 17.0.0.0. RDP enabled routers RS-1 and RS-2 pick up these packets on their RDP interfaces 1/1 and 1/2 and respond with router advertisement ICMP messages. RS-1 and RS-2 also periodically send out router advertisements on their RDP interfaces.

RDP Interfaces

An RDP interface is created by enabling RDP on an IP router interface. Once enabled, the RDP interface becomes active and joins the all-routers IP multicast group (224.0.0.2). The interface then transmits 3 initial router advertisement messages at random intervals that are no greater than 16 seconds apart. This process occurs upon activation to increase the likelihood that end hosts will quickly discover this router.

After an RDP interface becomes active and transmits its initial advertisements, subsequent advertisements are transmitted at random intervals that fall between a configurable range of time. This range of time is defined by specifying a maximum and minimum advertisement interval value. See "Defining the Advertisement Interval" on page 16-9 for more information. Because advertisements are transmitted at random intervals, the risk of system overload is reduced as advertisements from other routers on the same link are not likely to transmit at the same time.

It is important to note that advertisements are only transmitted on RDP interfaces if the following conditions are met:

- The RDP global status is enabled on the switch.
- An IP router interface exists and is in the enabled state.
- An RDP interface exists and is in the enabled state.
- Either VRRP is disabled or if VRRP is enabled, there is one or more Master IP addresses for the VLAN. If VRRP is enabled and if there are no Masters IP addresses, router advertisements are not sent on the VLAN. (See Chapter 18, "Configuring VRRP," for more information.)

The router advertisement is a multicast packet sent to the all-systems IP multicast group (224.0.0.1) or the broadcast address. If VRRP is enabled, the message should be filled with IP addresses obtained from VRRP Master IP address list; otherwise the IP address of the IP router interface is used.

Note that RDP is not recommended for detecting neighboring router failures, referred to as black holes, in the network. However, it is possible to use RDP as a supplement for black hole detection by setting RDP interface advertisement interval and lifetime values to values lower than the default values for these parameters. See "Defining the Advertisement Interval" on page 16-9 and "Setting the Advertisement Lifetime" on page 16-10 for more information.

Security Concerns

ICMP RDP packets are not authenticated, which makes them vulnerable to the following attacks:

- **Passive monitoring**—Attackers can use RDP to re-route traffic from vulnerable systems through the attacker's system. This allows the attacker to monitor or record one side of the conversation. However, the attacker must reside on the same network as the victim for this scenario to work.
- **Man in the middle**—Attacker modifies any of the outgoing traffic or plays man in the middle, acting as a proxy between the router and the end host. In this case, the victim thinks that it is communicating with an end host, not an attacker system. The end host thinks that is it communicating with a router because the attacker system is passing information through to the host from the router. If the victim is a secure web server that uses SSL, the attacker sitting in between the server and an end host could intercept unencrypted traffic. As is the case with passive monitoring, the attacker must reside on the same network as the victim for this scenario to work.
- **Denial of service (DoS)**—Remote attackers can spoof these ICMP packets and remotely add bad default-route entries into a victim's routing table. This would cause the victim to forward frames to the wrong address, thus making it impossible for the victim's traffic to reach other networks. Because of the large number of vulnerable systems and the fact that this attack will penetrate firewalls that do not stop incoming ICMP packets, this DoS attack can become quite severe. (See Chapter 13, "Configuring IP," and Chapter 24, "Configuring QoS," for more information about DoS attacks.)

Note. Security concerns associated with using RDP are generic to the feature as defined in RFC 1256 and not specific to this implementation.

Enabling/Disabling RDP

RDP is included in the base software and is available when the switch starts up. However, by default this feature is not operational until it is enabled on the switch.

To enable RDP operation on the switch, use the following command:

```
-> ip router-discovery enable
```

Once enabled, any existing RDP interfaces on the switch that are also enabled will activate and start to send initial advertisements. See "RDP Interfaces" on page 16-6 for more information.

To disable RDP operation on the switch, use the following command:

```
-> ip router-discovery disable
```

Use the **show ip router-discovery** command to determine the current operational status of RDP on the switch.

Creating an RDP Interface

An RDP interface is created by enabling RDP for an existing IP router interface, which is then advertised by RDP as an active router on the local network. Note that an RDP interface is not active unless RDP is also enabled for the switch.

To create an RDP interface, enter **ip router-discovery interface** followed by the name of the IP router interface then **enable**. For example, the following command creates an RDP interface for the IP router interface named Marketing:

```
-> ip router-discovery interface Marketing enable
```

The IP router interface name is the name assigned to the interface when it was first created. For more information about creating IP router interfaces, see Chapter 13, "Configuring IP."

The first time an RDP interface is enabled, it is not necessary to enter **enable** as part of the command. However, if the interface is subsequently disabled, then entering **enable** is required the next time this command is used. For example, the following sequence of commands initially enables an RDP interface for the Marketing IP router interface then disables and again enables the same interface:

```
-> ip router-discovery interface Marketing
-> ip router-discovery interface Marketing disable
-> ip router-discovery interface Marketing enable
```

When the above RDP interface becomes active, advertisement packets are transmitted on all active ports that belong to the VLAN associated with the Marketing interface. These packets contain the IP address associated with the Marketing interface for the purposes of advertising this interface on the network.

When an RDP interface is created, it is automatically configured with the following default parameter values:

RDP Interface Parameter	Default
Advertisement destination address.	All systems multicast (224.0.0.1)

Advertisement time interval defined by maximum and minimum values.	Maximum = 600 seconds
	Minimum = 450 seconds (0.75 * maximum value)

Advertisement lifetime.	1800 seconds (3 * maximum value)
Router IP address preference level.	0

It is only necessary to change the above parameter values if the default value is not sufficient. The following subsections provide information about how to configure RDP interface parameters if it is necessary to use a different value.

Specifying an Advertisement Destination Address

Active RDP interfaces transmit advertisement packets at random intervals and in response to ICMP solicitation messages received from network hosts. These packets are sent to one of two supported destination addresses: all systems multicast (224.0.0.1) or broadcast (255.255.255.255).

By default, RDP interfaces are configured to use the 224.0.0.1 as the destination address. To change the RDP destination address, use the **ip router-discovery interface advertisement-address** command. For example, the following command changes the destination address to the broadcast address:

```
-> ip router-discovery interface Marketing advertisement-address broadcast
```

Enter **all-systems-multicast** when using this command to change the destination address to 224.0.0.1. For example:

```
-> ip router-discovery interface Marketing advertisement-address all-systems-
multicast
```

Defining the Advertisement Interval

The advertisement interval represents a range of time, in seconds, in which RDP will transmit advertisement packets at random intervals. This range is defined by configuring a maximum amount of time that RDP will not exceed before the next transmission and configuring a minimum amount of time that RDP will observe before sending the next transmission. Both of these values are referred to as the maximum advertisement interval and the minimum advertisement interval.

Note that when an RDP interface becomes active, it transmits 3 advertisement packets at intervals no greater than 16 seconds. This facilitates a quick discovery of this router on the network. After these initial transmissions, advertisements occur at random times within the advertisement interval value or in response to solicitation messages received from network hosts.

Setting the Maximum Advertisement Interval

To set the maximum amount of time, in seconds, that RDP will allow between advertisements, use the **ip router-discovery interface max-advertisement-interval** command. For example, the following command sets this value to 1500 seconds for the Marketing IP router interface:

```
-> ip router-discovery interface Marketing max-advertisement-interval 1500
```

Make sure that the value specified with this command is *greater* than the current minimum advertisement interval value. By default, this value is set to 600 seconds.

Setting the Minimum Advertisement Interval

To set the minimum amount of time, in seconds, that RDP will allow between advertisements, use the **ip router-discovery interface min-advertisement-interval** command. For example, the following command sets this value to 500 seconds for the Marketing IP router interface:

```
-> ip router-discovery interface Marketing min-advertisement-interval 500
```

Make sure that the value specified with this command is *less* than the current maximum advertisement interval value. By default, this value is set to 0.75 * default maximum interval value (450 seconds if the maximum interval is set to its default value of 600 seconds).

Setting the Advertisement Lifetime

The advertisement lifetime value indicates how long, in seconds, the router IP address contained in an advertisement packet is considered valid by a host. This value is entered into the lifetime field of an advertisement packet so that it is available to hosts that receive these types of packets.

If a host does not receive another packet from the same router before the lifetime value expires, it assumes the router is no longer available and will drop the router IP address from its table. As a result, it is important that the lifetime value is always *greater* than the current maximum advertisement interval to ensure router transmissions occur before the lifetime value expires.

To set the advertisement lifetime value for packets transmitted from a specific RDP interface, use the **ip router-discovery interface advertisement-lifetime** command. For example, the following command sets this value to 3000 seconds for RDP packets sent from the Marketing IP router interface:

```
-> ip router-discovery interface Marketing advertisement-lifetime 3000
```

By default, the lifetime value is set to 3 * the current maximum interval value (1800 seconds if the maximum interval is set to its default value of 600 seconds).

Setting the Preference Levels for Router IP Addresses

A preference level is assigned to each router IP address contained within an advertisement packet. Hosts will select the IP address with this highest preference level to use as the default router gateway address. By default, this value is set to zero.

To specify a preference level for IP addresses advertised from a specific RDP interface, use the **ip router-discovery interface preference-level** command. For example, the following command sets this value to 10 for the IP address associated with the Marketing IP router interface:

```
-> ip router-discovery interface Marketing preference-level 10
```

Note that router IP address preference levels are only compared with the preference levels of other routers that exist on the same subnet. Set preference levels low to discourage selection of a specific router.

Verifying the RDP Configuration

To display information about the RDP configuration on the switch, use the **show** commands listed below:

show ip router-discovery Displays the current operational status of RDP on the switch. Also includes the number of advertisement packets transmitted and the number of solicitation packets received by all RDP interfaces on the switch.

show ip router-discovery interface Displays the current RDP status, related parameter values, and RDP traffic statistics for one or more switch router RDP interfaces.

For more information about the resulting displays from these commands, see the *OmniSwitch CLI Reference Guide*. An example of the output for the **show ip router-discovery** and **show ip router-discovery interface** commands is also given in "Quick Steps for Configuring RDP" on page 16-3.

17 Configuring DHCP Relay

The DHCP Relay allows you to use nonroutable protocols, such as the User Datagram Protocol (UDP), in a routing environment. UDP is a connectionless transport protocol that runs on top of IP networks and is used for applications that do not require the establishment of a session and end-to-end error checking. Email and file transfer are two applications that could use UDP. UDP offers a direct way to send and receive datagrams over an IP network and is primarily used for broadcasting messages. DHCP Relay allows UDP broadcast packets to be forwarded across VLANs that have IP routing enabled.

In This Chapter

This chapter describes the basic components of UDP Relay and how to configure them. CLI commands are used in the configuration examples. For more details about the syntax of commands, see the *OmniSwitch CLI Reference Guide*.

Information provided in this chapter includes:

- Quick steps for configuring DHCP Relay on page 17-4.
- Setting the IP address for Global DHCP on page 17-9.
- Setting the IP address for Global DHCP on page 17-9.
- Identifying the VLAN for Per-VLAN DHCP on page 17-9.
- Setting the Forward Delay time on page 17-10.
- Setting the Maximum Hops value on page 17-11.
- Setting the Relay Forwarding Option to Standard, Per-VLAN, or AVLAN on page 17-11.
- Using automatic IP configuration to obtain an IP address for the switch on page 17-12.
- Configuring relay for generic UDP service ports on page 17-13.
- Using the Relay Agent Information Option (Option-82) on page 17-15.
- Using DHCP Snooping on page 17-17.

For information about the IP protocol, see Chapter 13, “Configuring IP.”

DHCP Relay Specifications

The following table lists specifications for the DHCP Relay.

RFCs Supported	0951–Bootstrap Protocol 1534–Interoperation Between DHCP and BOOTP 1541–Dynamic Host Configuration Protocol 1542–Clarifications and Extensions for the Bootstrap Protocol 2132–DHCP Options and BOOTP Vendor Extensions, 1997 3046–DHCP Relay Agent Information Option, 2001
DHCP Relay Implementation	Global DHCP Per-VLAN DHCP AVLAN DHCP
DHCP Relay Service	BOOTP/DHCP (Bootstrap Protocol/Dynamic Host Configuration Protocol)
UDP Port Numbers	67 for Request 68 for Response
IP address allocation mechanisms	**Automatic**–DHCP assigns a permanent IP address to a host. **Dynamic**–DHCP assigns an IP address to a host for a limited period of time (or until the host explicitly relinquishes the address). **Manual**–The network administrator assigns a host's IP address and the DHCP conveys the address assigned by the host.
IP addresses supported for each Relay Service.	Maximum of 256 IP addresses for each Relay Service.
IP addresses supported for the Per-VLAN service	Maximum of 8 IP addresses for each VLAN relay service. Maximum of 256 VLAN relay services.
Maximum number of UDP relay services allowed per switch.	3
Maximum number of VLANs to which forwarded UDP service port traffic is allowed.	256
Maximum number of DHCP Snooping VLANs	64

DHCP Relay Defaults

The following table describes the default values of the DHCP Relay parameters.

Parameter Description	Command	Default Value/Comments
Default UDP service.	**ip udp relay**	BOOTP/DHCP
Forward delay time value for DHCP Relay	**ip helper forward delay**	3 seconds
Maximum number of hops.	**ip helper maximum hops**	4 hops
Packet forwarding option	**ip helper standard** **ip helper avlan only** **ip helper per-vlan only**	Standard
Automatic switch IP configuration for default VLAN 1.	**ip helper boot-up**	Disabled
Automatic switch IP configuration packet type (BootP or DHCP)	**ip helper boot-up enable**	BootP
Relay Agent Information Option	**ip helper agent-information**	Disabled
Switch-level DHCP Snooping	**ip helper dhcp-snooping**	Disabled
VLAN-level DHCP Snooping	**ip helper dhcp-snooping vlan**	Disabled

Quick Steps for Setting Up DHCP Relay

You should configure DHCP Relay on switches where packets are routed between IP networks.

There is no separate command for enabling or disabling the relay service. DHCP Relay is automatically enabled on the switch whenever a DHCP server IP address is defined. To set up DHCP Relay, proceed as follows:

1 Identify the IP address of the DHCP server. Where the DHCP server has IP address 128.100.16.1, use the following command:

```
-> ip helper address 128.100.16.1
```

2 Set the forward delay timer for the BOOTP/DHCP relay. To set the timer for a 15 second delay, use the following command:

```
-> ip helper forward delay 15
```

3 Set the maximum hop count value. To set a hop count of 3, use the following command:

```
-> ip helper maximum hops 3
```

Note. Optional. To verify the DHCP Relay configuration, enter the **show ip helper** command. The display shown for the DHCP Relay configured in the above Quick Steps is shown here:

```
-> show ip helper
Forward Delay (seconds) = 15
Max number of hops     = 3
Forward option     = standard
Forwarding Address:
128.100.16.1
```

For more information about this display, see the *OmniSwitch CLI Reference Guide*.

DHCP Relay Overview

The DHCP Relay service, its corresponding port numbers, and configurable options are as follows:

- DHCP Relay Service: BOOTP/DHCP
- UDP Port Numbers 67/68 for Request/Response
- Configurable options: DHCP server IP address, Forward Delay, Maximum Hops, Forwarding Option, automatic switch IP configuration

The port numbers indicate the destination port numbers in the UDP header. The DHCP Relay will verify that the forward delay time (specified by the user) has elapsed before sending the packet down to UDP with the destination IP address replaced by the address (also specified by the user).

If the relay is configured with multiple IP addresses, then the packet will be sent to all IP address destinations. The DHCP Relay also verifies that the maximum hop count has not been exceeded. If the forward delay time is *not* met or the maximum hop count is exceeded, the BOOTP/DHCP packet will be discarded by the DHCP Relay.

The forwarding option allows you to specify if the relay should operate in the standard, per-VLAN only, or AVLAN-only mode. The standard mode forwards all DHCP packets on a global relay service. The per-VLAN only mode forwards DHCP packets that originate from a specific VLAN. The AVLAN-only mode only forwards packets received on authenticated ports from non-authenticated clients. See "Setting the Relay Forwarding Option" on page 17-11 for more information.

An additional function provided by the DHCP Relay service enables automatic IP address configuration for default VLAN 1 when an unconfigured switch boots up. If this function is enabled, the switch broadcasts a BootP or a DHCP request packet at boot time. When the switch receives an IP address from a BootP/DHCP server, the address is assigned to default VLAN 1. See "Enabling Automatic IP Configuration" on page 17-12 for more information.

Alternately the relay function may be provided by an external router connected to the switch; in this case, the relay would be configured on the external router.

DHCP

DHCP (Dynamic Host Configuration Protocol) provides a framework for passing configuration information to Internet hosts on a TCP/IP network. It is based on the Bootstrap Protocol (BOOTP), adding the ability to automatically allocate reusable network addresses and additional configuration options. DHCP consists of the following two components:

- A protocol for delivering host-specific configuration parameters from a DHCP server to a host.
- A mechanism for allocating network addresses to hosts.

DHCP is built on a client-server model in which a designated DHCP server allocates network addresses and delivers configuration parameters to dynamically configured hosts. It supports the following three mechanisms for IP address allocation.

Automatic—DHCP assigns a permanent IP address to a host.

Dynamic—DHCP assigns an IP address to a host for a limited period of time (or until the host explicitly relinquishes the address).

Manual—The network administrator assigns a host's IP address and DHCP simply conveys the assigned address to the host.

DHCP and the OmniSwitch

The unique characteristics of the DHCP protocol require a good plan before setting up the switch in a DHCP environment. Since DHCP clients initially have no IP address, placement of these clients in a VLAN is hard to determine. In simple networks (e.g., one VLAN) rules do not need to be deployed to support the BOOTP/DHCP relay functionality.

In multiple VLAN network configurations, VLAN rules can be deployed to strategically support the processing and relay of DHCP packets. The most commonly used rules for this function are IP protocol rules, IP network address rules, and DHCP rules. All of these classify packets received on mobile ports based on the packet protocol type, source IP address, or if the packet is a DHCP request. See Chapter 8, "Defining VLAN Rules," for more information.

DHCP Relay and Authentication

Authentication clients may use DHCP to get an IP address. For Telnet authentication clients, an IP address is required for authentication. The DHCP server may be located in the default VLAN, an authenticated VLAN, or both. If authentication clients will be getting an IP address from a DHCP server located in an authenticated VLAN, DHCP relay can handle DHCP requests/responses for these clients as well.

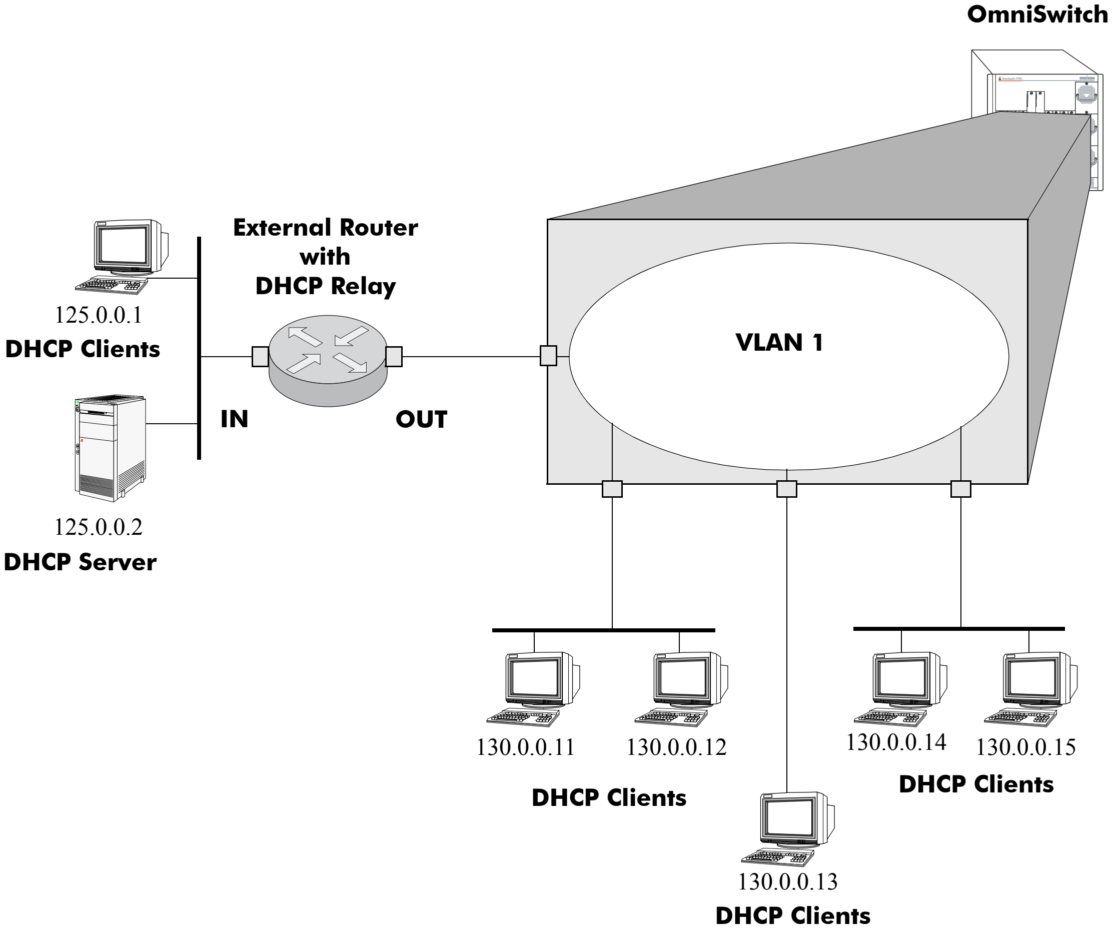
There are three relay forwarding options: standard, AVLAN only, and per-VLAN. All three support DHCP traffic to/from authenticated clients. However, the AVLAN only option specifies that only DHCP packets received on authenticated ports are processed. See "Setting the Relay Forwarding Option" on page 17-11 for more information.

Using DHCP Relay with authenticated VLANs and clients also requires relay configuration of the router port address of the authenticated VLAN. See Chapter 21, "Configuring Authenticated VLANs," for more information about this procedure.

External DHCP Relay Application

The DHCP Relay may be configured on a router that is external to the switch. In this application example the switched network has a single VLAN configured with multiple segments. All of the network hosts are DHCP-ready, meaning they obtain their network address from the DHCP server. The DHCP server resides behind an external network router, which supports the DHCP Relay functionality.

One requirement for routing DHCP frames is that the router must support DHCP Relay functionality to be able to forward DHCP frames. In this example, DHCP Relay is supported within an external router, which forwards request frames from the incoming router port to the outgoing router port attached to the OmniSwitch.



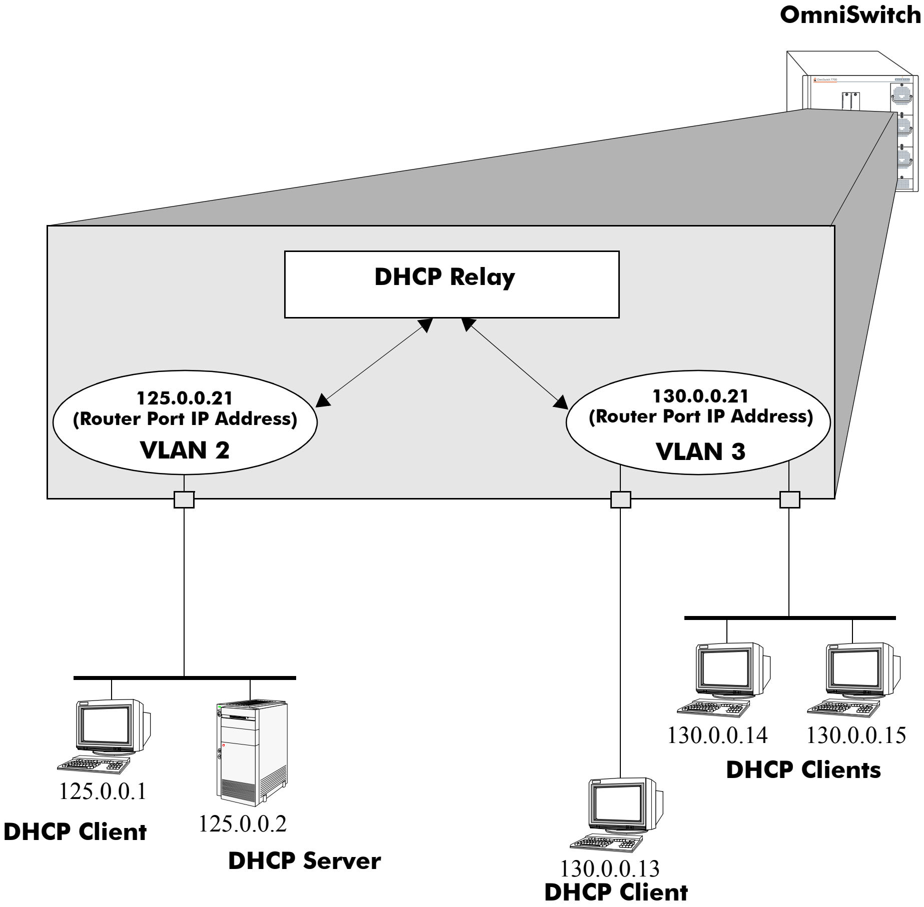
DHCP Clients are Members of the Same VLAN

The external router inserts the subnet address of the first hop segment into the DHCP request frames from the DHCP clients. This subnet address allows the DHCP server to locate the segment on which the requesting client resides. In this example, all clients attached to the OmniSwitch are DHCP-ready and will have the same subnet address (130.0.0.0) inserted into each of the requests by the router’s DHCP Relay function. The DHCP server will assign a different IP address to each of the clients. The switch does not need an IP address assigned and all DHCP clients will be members of either a default VLAN or an IP protocol VLAN.

Internal DHCP Relay

The internal DHCP Relay is configured using the UDP forwarding feature in the switch, available through the **ip helper address** command. For more information, see "DHCP Relay Implementation" on page 17-9.

This application example shows a network with two VLANs, each with multiple segments. All network clients are DHCP-ready and the DHCP server resides on just one of the VLANs. This example is much like the first application example, except that the DHCP Relay function is configured inside the switch.



DHCP Clients in Two VLANs

During initialization, each network client forwards a DHCP request frame to the DHCP server using the local broadcast address. For those locally attached stations, the frame will simply be switched.

In this case, the DHCP server and clients must be members of the same VLAN (they could also all be members of the default VLAN). One way to accomplish this is to use DHCP rules in combination with IP protocol rules to place all IP frames in the same VLAN. See Chapter 8, "Defining VLAN Rules," for more information.

Because the clients in the application example are not members of the same VLAN as the DHCP server, they must request an IP address via the DHCP Relay routing entity in the switch. When a DHCP request frame is received by the DHCP Relay entity, it will be forwarded from VLAN 3 to VLAN 2. All the DHCP-ready clients in VLAN 3 must be members of the same VLAN, and the switch must have the DHCP Relay function configured.

DHCP Relay Implementation

The OmniSwitch allows you to configure the DHCP Relay feature in one of two ways. You can set up a global DHCP request or you can set up the DHCP Relay based on the VLAN of the DHCP request. Both of these choices provide the same configuration options and capabilities. However, they are mutually exclusive. The following matrix summarizes the options.

Per-VLAN DHCP Relay	Global DHCP Relay	Effect
Disabled	Disabled	DHCP Request is flooded within its VLAN
Disabled	Enabled	DHCP Request is relayed to the Global Relay
Enabled	Disabled	DHCP Request is relayed to the Per-VLAN Relay
Enabled	Enabled	N/A

Global DHCP

For the global DHCP service, you must identify an IP address for the DHCP server.

Setting the IP Address

The DHCP Relay is automatically enabled on a switch whenever a DHCP server IP address is defined by using the **ip helper address** command. There is no separate command for enabling or disabling the relay service. You should configure DHCP Relay on switches where packets are routed between IP networks. The following command defines a DHCP server address:

```
-> ip helper address 125.255.17.11
```

The DHCP Relay forwards BOOTP/DHCP broadcasts to and from the specified address. If multiple DHCP servers are used, one IP address must be configured for each server. You can configure up to 256 addresses for each relay service.

To delete an IP address, use the **no** form of the **ip helper address** command. The IP address specified with this syntax will be deleted. If an IP address is not specified with this syntax, then *all* IP helper addresses are deleted. The following command deletes an IP helper address:

```
-> ip helper no address 125.255.17.11
```

Per-VLAN DHCP

For the Per-VLAN DHCP service, you must identify the number of the VLAN that makes the relay request.

Identifying the VLAN

You may enter one or more server IP addresses to which packets will be sent from a specified VLAN. Do this by using the **ip helper address vlan** command. The following syntax will identify the IP address 125.255.17.11 as the DHCP server for VLAN 3.

```
-> ip helper address 125.255.17.11 vlan 3
```

The following syntax identifies two DHCP servers for VLAN 4 at two different IP addresses.

```
-> ip helper address 125.255.17.11 125.255.18.11 vlan 4
```

To delete an IP address, use the **no** form of the **ip helper address** command. The IP address specified with this syntax will be deleted. If an IP address is not specified with this syntax, then *all* IP helper addresses are deleted. The following command deletes an helper address for IP address 125.255.17.11:

```
-> ip helper no address 125.255.17.11
```

The following command deletes all IP helper addresses:

```
-> ip helper no address
```

Configuring BOOTP/DHCP Relay Parameters

Once the IP address of the DHCP server(s) is defined and the DHCP Relay is configured for either Global DHCP request or Per-VLAN DHCP request, you can set the following optional parameter values to configure BOOTP relay.

- The forward delay time.
- The hop count.
- The relay forwarding option.

The only parameter that is required for BOOTP relay is the IP address to the DHCP server or to the next hop to the DHCP server. The default values can be accepted for forward delay, hop count, and relay forwarding option.

Alternately the relay function may be provided by an external router connected to the switch; in this case, the relay would be configured on the external router.

Setting the Forward Delay

Forward Delay is a time period that gives the local server a chance to respond to a client before the relay forwards it further out in the network.

The UDP packet that the client sends contains the elapsed boot time. This is the amount of time, measured in seconds, since the client last booted. DHCP Relay will not process the packet unless the client’s elapsed boot time value is equal to or greater than the configured value of the forward delay time. If a packet contains an elapsed boot time value that is less than the specified forward delay time value, DHCP Relay discards the packet.

The forward delay time value applies to all defined IP helper addresses. The following command sets the forward delay value of 10 seconds.

```
-> ip helper forward delay 10
```

The range for the forward delay time value is 1 to 65535 seconds.

Setting Maximum Hops

This value specifies the maximum number of relays the BOOTP/DHCP packet can go through until it reaches its server destination. This limit keeps packets from “looping” through the network. If a UDP packet contains a hop count equal to the hops value, DHCP Relay discards the packet. The following syntax is used to set a maximum of four hops.

```
-> ip helper maximum hops 4
```

The hops value represents the maximum number of relays. The range is from one to 16 hops. The default maximum hops value is set to four. This maximum hops value only applies to DHCP Relay. All other switch services will ignore this value.

Setting the Relay Forwarding Option

This value specifies if DHCP Relay should operate in a standard, AVLAN, or per-VLAN only forwarding mode. If the AVLAN only option is selected, only DHCP packets received on authenticated ports are processed. By default, the forwarding option is set to standard. To change the forwarding option value, enter **ip helper** followed by **standard**, **avlan only**, or
per-vlan only. For example,

```
-> ip helper avlan only
-> ip helper standard
-> ip helper per-vlan only
```

Using Automatic IP Configuration

An additional function of the DHCP Relay feature enables a switch to broadcast a BootP or DHCP request packet at boot time to obtain an IP address for default VLAN 1. This function is separate from the previously described functions (such as global DHCP, per-VLAN DHCP and related configurable options) in that enabling or disabling automatic IP configuration does not exclude or prevent other DHCP Relay functionality.

Note. Automatic IP address configuration only supports the assignment of a *permanent* IP address to the switch. Make sure that the DHCP server is configured with such an address before using this feature.

Using automatic IP configuration also allows the switch to specify the type of request packet to send; BootP (the default) or DHCP. When the BootP/DHCP server receives the request packet from the switch, it processes the request and sends an appropriate reply packet. When the switch receives a reply packet from the BootP/DHCP server, one or more of the following occurs:

- The router port for VLAN 1 is assigned the IP address provided by the server.
- If the reply packet contains a subnet mask for the IP address, the mask is applied to the VLAN 1 router port address. Otherwise, a default mask is determined based upon the class of the IP address. For example, if the IP address is a Class A, B, or C address, then 255.0.0.0, 255.255.0.0, or 255.255.255.0 is used for the subnet mask.
- If the reply packet from the server contains a gateway IP address, then a static route entry of 0.0.0.0 is created on the switch with the gateway address provided by the server.

Note. If the VLAN 1 router port is already configured with an IP address, the switch does not broadcast a request packet at boot time even if automatic IP configuration is enabled.

To verify IP router port configuration for VLAN 1, use the **show ip interface** and **show ip route** commands. For more information about these commands, refer to the *OmniSwitch CLI Reference Guide*.

Enabling Automatic IP Configuration

By default, this function is disabled on the switch. To enable automatic IP configuration and specify the type of request packet, use the **ip helper boot-up** command. For example:

```
-> ip helper boot-up enable DHCP
-> ip helper boot-up enable BOOTP
```

Once enabled, the next time the switch boots up, DHCP Relay will broadcast a BootP (the default) or DHCP request packet to obtain an IP address for default VLAN 1.

To disable automatic IP configuration for the switch, use the **ip helper boot-up** command with the **disable** option, as shown below:

```
-> ip helper boot-up disable
```

Configuring UDP Port Relay

In addition to configuring a relay operation for BOOTP/DHCP traffic on the switch, it is also possible to configure relay for generic UDP service ports (i.e., NBNS/NBDD, other well-known UDP service ports, and service ports that are not well-known). This is done using UDP Port Relay commands to enable relay on these types of ports and to specify up to 256 VLANs that can forward traffic destined for these ports.

The UDP Port Relay function is separate from the previously described functions (such as global DHCP, per-VLAN DHCP, and automatic IP configuration) in that using UDP Port Relay does not exclude or prevent other DHCP Relay functionality. However, the following information is important to remember when configuring BOOTP/DHCP relay and UDP port relay:

- UDP port relay supports up to three UDP relay services at any one time and in any combination.

Note. If the relay service for BOOTP/DHCP is disabled when the switch reboots, the service is automatically enabled when the switch comes back up. If there were three non-BOOTP/DHCP relay services already enabled before the reboot, the most recent service enabled is disabled and replaced with the BOOTP/DHCP relay service.

- The **ip helper** commands are used to configure BOOTP/DHCP relay and the **ip udp port** commands are used to configure UDP port relay. The **ip udp relay** command, however, is also used to enable or disable relay for BOOTP/DHCP well known ports 67 and 68.
- If the BOOTP/DHCP relay service is disabled, the **ip helper** configuration is *not* retained and all dependant functionality (i.e., automatic IP configuration for VLAN 1, Telnet and HTTP client authentication, etc.) is disrupted.
- Relaying BOOTP/DHCP traffic is available on a global and per-VLAN basis. Using this function on a per-VLAN basis requires setting the DHCP relay forwarding mode to **per-vlan only**. UDP port relay for generic services is only available on a per-VLAN basis, but does not require enabling the **per-vlan only** forwarding option.

Configuring UDP Port Relay for generic UDP services is a two-step process. The first step involves enabling UDP Port Relay on the generic service port. The second step involves specifying a VLAN that relay will forward traffic destined for the generic service port. Both steps are required and are described below.

Enabling/Disabling UDP Port Relay

By default, a global relay operation is enabled for BOOTP/DHCP relay well-known ports 67 and 68, which becomes active when an IP network host address for a DHCP server is specified. To enable or disable a relay operation for a UDP service port, use the **ip udp relay** command. For example, the following command enables relay on the DNS well-known service port:

```
-> ip udp relay DNS
```

To enable relay on a user-defined (not well-known) UDP service port, then enter the service port number instead of the service name. For example, the following command enables relay on service port 3047:

```
-> ip udp relay 3047
```

To disable a relay operation for a UDP service port, use the **no** form of the **ip udp relay** command. For example, the following command disables relay on the DNS well-known service port:

```
-> no ip udp relay dns
```

For more information about using the **ip udp relay** command, see the *OmniSwitch CLI Reference Guide*.

Specifying a Forwarding VLAN

To specify which VLAN(s) UDP Port Relay will forward traffic destined for a generic UDP service port, use the **ip udp relay vlan** command. For example, the following command assigns VLAN 5 as a forwarding VLAN for the DNS well-known service port:

```
-> ip udp relay dns vlan 5
```

Note that the **ip udp relay vlan** command only works if UDP Port Relay is already enabled on the specified service port. In addition, when assigning a VLAN to the BOOTP/DHCP service ports, set the DHCP relay forwarding mode to **per-vlan only** first before trying to assign the VLAN.

It is also possible to assign up to 256 forwarding VLANs to each generic service port. To specify more than one VLAN with a single command, enter a range of VLANs. For example, the following command assigns VLANs 6 through 8 and VLAN 10 as forwarding VLANs for the NBNS/NBDD well-known service ports:

```
-> ip udp relay nbnsnbdd vlan 6-8 10
```

If UDP Port Relay was enabled on a not well-known service port, then enter the service port number instead of the service name. For example, the following command assigns VLAN 100 as a forwarding VLAN for UDP service port 3047:

```
-> ip udp relay 3047 vlan 100
```

To remove a VLAN association with a UDP service port, use the **no** form of the **ip udp relay vlan** command. For example, the following command removes the VLAN 6 association with the NBNS/NBDD well-known service port:

```
-> no ip udp relay nbnsnbdd vlan 6
```

For more information about using the **ip udp relay vlan** command, see the *OmniSwitch CLI Reference Guide*.

Configuring DHCP Security Features

There are two DHCP security features available: DHCP relay agent information option (Option-82) and DHCP Snooping. The DHCP Option-82 feature enables the relay agent to insert identifying information into client-originated DHCP packets before the packets are forwarded to the DHCP server. The DHCP Snooping feature filters DHCP packets between untrusted sources and a trusted DHCP server and builds a binding database to log DHCP client information.

Although DHCP Option-82 is a subcomponent of DHCP Snooping, these two features are mutually exclusive. If the DHCP Option-82 feature is enabled for the switch, then DHCP Snooping is not available. The reverse is also true; if DHCP Snooping is enabled, then DHCP Option-82 is not available. In addition, the following differences exist between these two features:

- DHCP Snooping does require and use the Option-82 data insertion capability, but does not implement any other behaviors defined in RFC 3046.
- DHCP Snooping will automatically drop client DHCP packets that already have Option-82 information present. The DHCP Option-82 feature provides configurable options for dealing with such packets.
- DHCP Snooping is configurable at the switch level and on a per-VLAN basis, but DHCP Option-82 is only configurable at the switch level.

The following sections provide additional information about each DHCP security feature and how to configure feature parameters using the Command Line Interface (CLI).

Using the Relay Agent Information Option (Option-82)

This implementation of the DHCP relay agent information option (Option-82) feature is based on the functionality defined in RFC 3046. By default DHCP Option-82 functionality is disabled. The **ip helper agent-information** command is used to enable this feature at the switch level.

When this feature is enabled, communications between a DHCP client and a DHCP server are authenticated by the relay agent. To accomplish this task, the agent adds Option-82 data to the end of the options field in DHCP packets sent from a client to a DHCP server. Option-82 consists of two suboptions: Circuit ID and Remote ID. The agent fills in the following information for each of these suboptions:

- **Circuit ID**—the VLAN ID and slot/port from where the DHCP packet originated.
- **Remote ID**—the MAC address of the router interface associated with the VLAN ID specified in the Circuit ID suboption.

The DHCP Option-82 feature is only applicable when DHCP relay is used to forward DHCP packets between clients and servers associated with different VLANs. In addition, a secure IP network must exist between the relay agent and the DHCP server.

How the Relay Agent Processes DHCP Packets from the Client

The following table describes how the relay agent processes DHCP packets received from clients when the Option-82 feature is enabled for the switch:

If the DHCP packet from the client ...	The relay agent ...
Contains a zero gateway IP address (0.0.0.0) and no Option-82 data.	Inserts Option-82 with unique information to identify the client source.
Contains a zero gateway IP address (0.0.0.0) and Option-82 data.	Drops the packet, keeps the Option-82 data and forwards the packet, or replaces the Option-82 data with its own Option-82 data and forwards the packet. The action performed by the relay agent in this case is determined by the agent information policy that is configured through the **ip helper agent-information policy** command. By default, this type of DHCP packet is dropped by the agent.
Contains a non-zero gateway IP address and no Option-82 data.	Drops the packet without any further processing.
Contains a non-zero gateway IP address and Option-82 data.	Drops the packet if the gateway IP address matches a local subnet, otherwise the packet is forwarded without inserting Option-82 data.

How the Relay Agent Processes DHCP Packets from the Server

Note that if a DHCP server does not support Option-82, the server strips the option from the packet. If the server does support this option, the server will retain the Option-82 data received and send it back in a reply packet.

When the relay agent receives a DHCP packet from the DHCP server and the Option-82 feature is enabled, the agent will:

1 Extract the VLAN ID from the Circuit ID suboption field in the packet and compare the MAC address of the IP router interface for that VLAN to the MAC address contained in the Remote ID suboption field in the same packet.

2 If the IP router interface MAC address and the Remote ID MAC address are not the same, then the agent will drop the packet.

3 If the two MAC addresses match, then a check is made to see if the slot/port value in the Circuit ID suboption field in the packet matches a port that is associated with the VLAN also identified in the Circuit ID suboption field.

4 If the slot/port information does not identify an actual port associated with the Circuit ID VLAN, then the agent will drop the packet.

5 If the slot/port information does identify an actual port associated with the Circuit ID VLAN, then the agent strips the Option-82 data from the packet and unicasts the packet to the port identified in the Circuit ID suboption.

Enabling the Relay Agent Information Option-82

Use the **ip helper agent-information** command to enable the DHCP Option-82 feature for the switch. For example:

```
-> ip helper agent-information enable
```

This same command is also used to disable this feature. For example:

```
-> ip helper agent-information disable
```

Note that because this feature is not available on a per-VLAN basis, DHCP Option-82 functionality is not restricted to ports associated with a specific VLAN. Instead, DHCP traffic received on all ports is eligible for Option-82 data insertion when it is relayed by the agent.

Configuring a Relay Agent Information Option-82 Policy

As previously mentioned, when the relay agent receives a DHCP packet from a client that already contains Option-82 data, the packet is dropped by default. However, it is possible to configure a DHCP Option-82 policy that directs the relay agent to drop, keep, or replace the existing Option-82 data and then forward the packet to the server.

To configure a DHCP Option-82 policy, use the **ip helper agent-information policy** command. The following parameters are available with this command to specify the policy action:

- **drop**—The DHCP packet is dropped (the default).
- **keep**—The existing Option-82 data in the DHCP packet is retained and the packet is forwarded to the server.
- **replace**—The existing Option-82 data in the DHCP packet is replaced with local relay agent data and then forwarded to the server.

For example, the following commands configure DHCP Option-82 policies:

```
-> ip helper agent-information policy drop

-> ip helper agent-information policy keep

-> ip helper agent-information policy replace
```

Note that this type of policy applies to all DHCP packets received on all switch ports. In addition, if a packet that contains existing Option-82 data also contains a gateway IP address that matches a local subnet address, the relay agent will drop the packet and not apply any existing Option-82 policy.

Using DHCP Snooping

Using DHCP Snooping improves network security by filtering DHCP messages received from devices outside the network and building and maintaining a binding table (database) to track access information for such devices.

In order to identify DHCP traffic that originates from outside the network, DHCP Snooping categorizes ports as either trusted or untrusted. A port is trusted if it is connected to a device inside the network, such as a DHCP server. A port is untrusted if it is connected to a device outside the network, such as a customer switch or workstation.

When DHCP Snooping is first enabled, all ports are considered untrusted. It is important to then configure ports connected to a DHCP server inside the network as a trusted port. See "Configuring the Port Trust Mode" on page 17-20 for more information.

If a DHCP packet is received on an untrusted port, then it is considered an untrusted packet. If a DHCP packet is received on a trusted port, then it is considered a trusted packet. DHCP Snooping only filters untrusted packets and will drop such packets if one or more of the following conditions are true:

- The packet received is a DHCP server packet, such as a DHCPOFFER, DHCPACK, or DHCPNAK packet. When a server packet is received on an untrusted port, DHCP Snooping knows that it is not from a trusted server and discards the packet.
- The source MAC address of the packet and the DHCP client hardware address contained in the packet are not the same address.
- The packet is a DHCPRELEASE or DHCPDECLINE broadcast message that contains a source MAC address found in the DHCP Snooping binding table, but the interface information in the binding table does not match the interface on which the message was received.
- The packet includes a relay agent IP address that is a non-zero value.
- The packet already contains Option-82 data in the options field.

If none of the above are true, then the relay agent accepts and forwards the packet. When the relay agent receives a DHCPACK packet from a server, the agent extracts the following information to create an entry in the DHCP Snooping binding table:

- MAC address of the DHCP client.
- IP address for the client that was assigned by the DHCP server.
- The port from where the DHCP packet originated.
- The VLAN associated with the port from where the DHCP packet originated.
- The lease time for the assigned IP address.
- The binding entry type; dynamic or static (user-configured).

After extracting the above information and populating the binding table, the agent then forwards the packet to the port from where the packet originated. Basically, the DHCP Snooping features prevents the normal flooding of DHCP traffic. Instead, packets are delivered only to the appropriate client and server ports.

Note that DHCP Snooping only applies to traffic that is relayed between VLANs. If a DHCP server and client reside within the same VLAN domain, then DHCP Snooping is not applied to communications between these devices.

DHCP Snooping Configuration Guidelines

Consider the following when configuring the DHCP Snooping feature:

- DHCP Snooping requires the use of the relay agent to process DHCP packets. As a result, DHCP clients and servers must reside in different VLANs so that the relay agent is engaged to forward packets between the VLAN domains. See "Configuring BOOTP/DHCP Relay Parameters" on page 17-10 for information about how to configure the relay agent on the switch.
- Configure ports connected to DHCP servers within the network as trusted ports. See "Configuring the Port Trust Mode" on page 17-20 for more information.

- Make sure that Option-82 data insertion is always enabled at the switch or VLAN level. See "Enabling DHCP Snooping" on page 17-19 for more information.
- The DHCP sever must support the Option-82 feature or at a minimum retain and echo back the Option-82 data field.

Enabling DHCP Snooping

There are two levels of operation available for the DHCP Snooping feature: switch level or VLAN level. These two levels are exclusive of each other in that they both can not operate on the switch at the same time. In addition, if the global DHCP relay agent information option (Option-82) is enabled for the switch, then DHCP Snooping at any level is not available. See "Using the Relay Agent Information Option (Option-82)" on page 17-15 for more information.

Note. DHCP Snooping drops server packets received on untrusted ports (ports that connect to devices outside the network or firewall). It is important to configure ports connected to DHCP servers as trusted ports so that traffic to/from the server is not dropped.

Switch-level DHCP Snooping

By default, DHCP Snooping is disabled for the switch. To enable this feature at the switch level, use the **ip helper dhcp-snooping** command. For example:

```
-> ip helper dhcp-snooping enable
```

When DHCP Snooping is enabled at the switch level, all DHCP packets received on all switch ports are screened/filtered by DHCP Snooping. By default, only client DHCP traffic is allowed on the ports, unless the trust mode for a port is configured to block or allow all DHCP traffic. See "Configuring the Port Trust Mode" on page 17-20 for more information.

In addition, the following functionality is also activated by default when DHCP Snooping is enabled:

- The DHCP Snooping binding table is created and maintained.
- MAC address verification is performed to compare the source MAC address of the DHCP packet with the client hardware address contained in the packet.
- Option-82 data is inserted into the packet and then DHCP reply packets are only sent to the port from where the DHCP request originated, instead of flooding these packets to all ports.

To enable or disable any of the above functionality at the switch level, use the following commands:.

ip helper dhcp-snooping binding
ip helper dhcp-snooping mac-address verification
ip helper dhcp-snooping option-82 data-insertion

Note the following when disabling DHCP Snooping functionality:

- Disabling Option-82 is not allowed if the binding table is enabled.
- Enabling the binding table is not allowed if Option-82 data insertion is not enabled at either the switch or VLAN level.

VLAN-Level DHCP Snooping

To enable DHCP Snooping at the VLAN level, use the **ip helper dhcp-snooping vlan** command. For example, the following command enables DHCP Snooping for VLAN 200:

```
-> ip helper dhcp-snooping vlan 200
```

When this feature is enabled at the VLAN level, DHCP Snooping functionality is only applied to ports that are associated with a VLAN that has this feature enabled. Up to 64 VLANs can have DHCP Snooping enabled. Note that enabling DHCP Snooping at the switch level is not allowed if it is enabled for one or more VLANs.

By default, when DHCP Snooping is enabled for a specific VLAN, MAC address verification and Option-82 data insertion is also enabled for the VLAN by default. To disable or enable either of these two features, use the **ip helper dhcp-snooping vlan** command with either the **mac-address verification** or **option-82 data-insertion** parameters. For example:

```
-> ip helper dhcp-snooping vlan 200 mac-address verification disable

-> ip helper dhcp-snooping vlan 200 option-82 data-insertion disable
```

Note that if the binding table functionality is enabled, disabling Option-82 data insertion for the VLAN is not allowed. See "Configuring the DHCP Snooping Binding Table" on page 17-21 for more information.

Note. If DHCP Snooping is *not* enabled for a VLAN, then all ports associated with the VLAN are considered trusted ports. VLAN-level DHCP Snooping does not filter DHCP traffic on ports associated with a VLAN that does not have this feature enabled.

Configuring the Port Trust Mode

The DHCP Snooping trust mode for a port determines whether or not the port accepts all DHCP traffic, client-only DHCP traffic, or blocks all DHCP traffic. The following trust modes for a port are configurable using the **ip helper dhcp-snooping port** command:

- **client-only**—The default mode applied to ports when DHCP Snooping is enabled. This mode restricts DHCP traffic on the port to only DHCP client-related traffic. When this mode is active for the port, the port is considered an untrusted interface.
- **trust**—This mode does not restrict DHCP traffic on the port. When this mode is active on a port, the port is considered a trusted interface. In this mode the port behaves as if DHCP Snooping is not enabled.
- **block**—This mode blocks all DHCP traffic on the port. When this mode is active for the port, the port is considered an untrusted interface.

To configure the trust mode for one or more ports, use the **ip helper dhcp-snooping port** command. For example, the following command changes the trust mode for port 1/12 to blocked:

```
-> ip helper dhcp-snooping port 1/12 block
```

It is also possible to specify a range of ports. For example, the following command changes the trust mode for ports 2/1 through 2/10 to trusted:

```
-> ip helper dhcp-snooping port 2/1-10 trust
```

Note it is necessary to configure ports that are connected to DHCP servers within the network and/or firewall as trusted ports so that necessary DHCP traffic to/from the server is not blocked. Configuring the port mode as trusted also identifies the device connected to that port as a trusted device within the network.

Configuring the DHCP Snooping Binding Table

The DHCP Snooping binding table is automatically enabled when DHCP Snooping is enabled at either the switch or VLAN level. This table is used by DHCP Snooping to filter DHCP traffic that is received on untrusted ports.

Entries are made in this table when the relay agent receives a DHCPACK packet from a trusted DHCP server. The agent extracts the client information, populates the binding table with the information and then forwards the DHCPACK packet to the port where the client request originated.

To enable or disable the DHCP Snooping binding table, use the **ip helper dhcp-snooping binding** command. For example:

```
-> ip helper dhcp-snooping binding enable
```

```
-> ip helper dhcp-snooping binding disable
```

Note that enabling the binding table functionality is not allowed if Option-82 data insertion is *not* enabled at either the switch or VLAN level.

In addition, it is also possible to configure static binding table entries. This type of entry is created using available **ip helper dhcp-snooping binding** command parameters to define the static entry. For example, the following command creates a static DHCP client entry:

```
-> ip helper dhcp-snooping binding 00:2a:95:51:6c:10 port 1/15 address
17.15.3.10 lease-time 3 vlan 200
```

To remove a static binding table entry, use the **no** form of the **ip helper dhcp-snooping binding** command. For example:

```
-> no ip helper dhcp-snooping binding 00:2a:95:51:6c:10 port 1/15 address
17.15.3.10 lease-time 3 vlan 200
```

To view the DHCP Snooping binding table contents, use the **show ip helper dhcp-snooping binding** command. See the *OmniSwitch CLI Reference Guide* for example outputs of this command.

Configuring the Binding Table Timeout

The contents of the DHCP Snooping binding table resides in the switch memory. In order to preserve table entries across switch reboots, the table contents is automatically saved to the **dhcpBinding.db** file located in the **/flash/switch** directory.

The amount of time, in seconds, between each automatic save is referred to as the binding table timeout value. By default, the timeout value is 300 seconds. To configure this value, use the **ip helper dhcp-snooping binding timeout** command. For example, the following command sets the timeout value to 600 seconds:

```
-> ip helper dhcp-snooping binding timeout 600
```

Each time an automatic save is performed, the **dhcpBinding.db** file is time stamped.

Synchronizing the Binding Table

To synchronize the contents of the **dhcpBinding.db** file with the binding table contents that resides in memory, use the **ip helper dhcp-snooping binding action** command. This command provides two parameters: **purge** and **renew**. Use the **purge** parameter to clear binding table entries in memory and the **renew** parameter to populate the binding table with the contents of the **dhcpBinding.db** file. For example:

```
-> ip helper dhcp-snooping binding action purge

-> ip helper dhcp-snooping binding action renew
```

Synchronizing the binding table is only done when this command is used. There is no automatic triggering of this function. In addition, it is important to note that synchronizing the binding table loads **dhcpBinding.db** file contents into memory. This is the reverse of saving the binding table contents in memory to the **dhcpBinding.db** file, which is done at automatic time intervals as defined by the binding table timeout value. See "Configuring the Binding Table Timeout" on page 17-21 for more information.

Verifying the UDP Relay Configuration

To display information about the UDP Relay configuration, use the **show** commands listed below.

For more information about the resulting displays from these commands, see the *OmniSwitch CLI Reference Guide*. An example of the output for the **show ip helper** command is also given in "Quick Steps for Setting Up DHCP Relay" on page 17-4.

show ip helper	Displays the current forward delay time, the maximum number of hops, the forwarding option (standard or AVLAN only), and each of the DHCP server IP addresses configured. Also displays the current configuration status for the DHCP relay agent information option (Option-82) and DHCP Snooping features.
show ip helper stats	Displays the number of packets the DHCP Relay service has received and transmitted, the number of packets dropped due to forward delay and maximum hops violations, and the number of packets processed since the last time these statistics were displayed.
show ip udp relay service	Displays current configuration for UDP services by service name or by service port number.
show ip udp relay statistics	Displays the current statistics for each UDP port relay service. These statistics include the name of the service, the forwarding VLAN(s) configured for that service, and the number of packets the service has sent and received.
show ip udp relay destination	Displays the VLAN assignments to which the traffic received on the specified UDP service port is forwarded.
show ip helper dhcp-snooping vlan	Displays a list of VLANs that have DHCP Snooping enabled and whether or not MAC address verification and Option-82 data insertion is enabled for each VLAN.
show ip helper dhcp-snooping port	Displays the DHCP Snooping trust mode for the port and the number of packets destined for the port that were dropped due to a DHCP Snooping violation.
show ip helper dhcp-snooping binding	Displays the contents of the DHCP Snooping binding table (database).

18 Configuring VRRP

The Virtual Router Redundancy Protocol (VRRP) is a standard router redundancy protocol supported in IP version 4. It is based on RFC 2338 and provides redundancy by eliminating the single point of failure inherent in a default route environment.

In This Chapter

This chapter describes VRRP and how to configure it through the Command Line Interface (CLI). CLI commands are used in the configuration examples; for more details about the syntax of commands, see the *OmniSwitch CLI Reference Guide*.

This chapter provides an overview of VRRP and includes information about the following:

- Virtual routers—see "Creating a Virtual Router" on page 18-8.
- IP addresses for virtual routers—see "Specifying an IP Address for a Virtual Router" on page 18-9.
- VRRP advertisement interval—see "Configuring the Advertisement Interval" on page 18-10.
- Virtual router priority—see "Configuring Virtual Router Priority" on page 18-10.
- Preempting virtual routers—see "Setting Preemption for Virtual Routers" on page 18-11.
- VRRP traps—see "Setting VRRP Traps" on page 18-12.
- VRRP tracking—see "Creating Tracking Policies" on page 18-13.
- Verifying the VRRP configuration—see "Verifying the VRRP Configuration" on page 18-14.

VRRP Specifications

RFCs Supported	RFC 2338–Virtual Router Redundancy Protocol RFC 2787–Definitions of Managed Objects for the Virtual Router Redundancy Protocol
Compatible with HSRP?	No
Maximum number of virtual routers	255
Maximum number of IP addresses	1 for the IP address owner; more than 1 address may be configured if the router is a backup for a master router that supports multiple addresses

VRRP Defaults

The following table lists the defaults for VRRP configuration through the **vrrp** command and the relevant command keywords:

Description	Keyword	Default
Virtual router enabled or disabled	**enable | disable | on | off**	Virtual routers are disabled (off).
Priority	**priority**	100
Preempt mode	**preempt | no preempt**	Preempt mode is enabled.
Advertising interval	**advertising] interval**	1 second

In addition, other defaults for VRRP include:

Description	Command	Default
VRRP traps	**vrrp trap**	Disabled
VRRP tracking	**vrrp track**	Enabled
VRRP delay	**vrrp delay**	45 seconds

Quick Steps for Creating a Virtual Router

1 Create a virtual router. Specify a virtual router ID (VRID) and a VLAN ID. For example:

```
-> vrrp 23 4
```

The VLAN must already be created on the switch. For information about creating VLANs, see Chapter 5, "Configuring VLANs."

2 Configure an IP address for the virtual router.

```
-> vrrp 23 4 ip 10.10.2.3
```

3 Repeat steps 1 and 2 on all of the physical switches that will participate in backing up the address(es) associated with the virtual router.

4 Enable VRRP on each switch.

```
-> vrrp 23 4 enable
```

Note. *Optional*. To verify the VRRP configuration, use the **show vrrp** command. For example:

```
-> show vrrp
VRRP trap generation: Enabled
VRRP startup delay: 75
               IP           Admin                           Adv
VRID  VLAN   Address(es)    Status    Priority  Preempt    Interval
----+ ----+ -------------+----------+----------+----------+----------
  1     1   192.168.170.1   Enabled     255      Yes       1
            192.168.170.2
  2    15   10.2.25.254     Disabled    100    None        No           1

-> show vrrp 1
Virtual Router VRID = 1 on VLAN = 1
   Admin Status       = Enabled
   Priority           = 255
   Preempt            = Yes
   Adv. Interval      = 1
   Virtual MAC        = 00-00-5E-00-01-01
   IP Address(es)
      192.168.170.1
      192.168.170.2
```

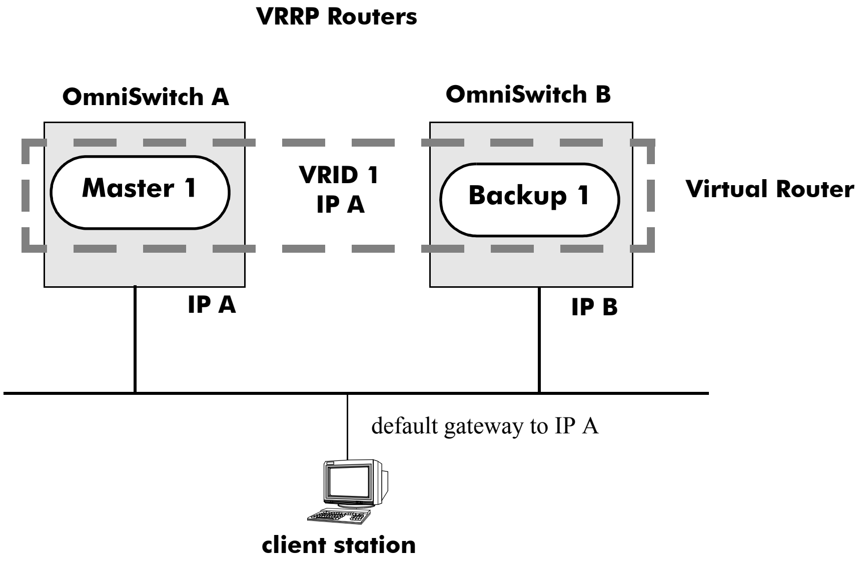
For more information about this and other **show** commands, see the *OmniSwitch CLI Reference Guide*.

VRRP Overview

VRRP allows routers on a LAN to back up a default route. VRRP dynamically assigns responsibility for a virtual router to a physical router (VRRP router) on the LAN. The virtual router is associated with an IP address (or set of IP addresses) on the LAN. A virtual router master is elected to forward packets for the virtual router's IP address. If the master router becomes unavailable, the highest priority backup router will transition to the master state.

Note. The IP address that is backed up may be the IP address of a physical router, or it may be a virtual IP address.

The example provided here is intended for understanding VRRP and does not show a configuration that would be used in an actual network.



VRRP Redundancy Example

In this example, each physical router is configured with a virtual router, VRID 1, which is associated with IP address A. OmniSwitch A is the master router because it contains the physical interface to which IP address A is assigned. OmniSwitch B is the backup router. The client is configured with a gateway address of IP A.

When VRRP is configured on these switches, and both switches are available, OmniSwitch A will respond to ARP requests for IP address A using the virtual router's MAC address (00:00:5E:00:01:01) instead of the physical MAC address assigned to the interface. OmniSwitch A will accept packets sent to the virtual MAC address and forward them as appropriate; it will also accept packets addressed to IP address A (such as ICMP ping requests).

OmniSwitch B will respond to ARP requests for IP address B using the interface's physical MAC address. It will not respond to ARP requests for IP address A or to the virtual router MAC address.

If OmniSwitch A becomes unavailable, OmniSwitch B becomes the master router. OmniSwitch B will then respond to ARP requests for IP address A using the virtual router's MAC address (00:00:5E:00:01:01). It will also forward packets for IP address B and respond to ARP requests for IP address B using the OmniSwitch's physical MAC address. OmniSwitch B, however, cannot accept packets addressed to IP address A (such as ICMP ping requests).

OmniSwitch B uses IP address B to access the LAN, but IP address B is not backed up. If OmniSwitch B becomes unavailable, IP address B is unavailable.

Why Use VRRP?

An end host may use dynamic routing or router discovery protocols to determine its first hop toward a particular IP destination. With dynamic routing, large timer values are required and may cause significant delay in the detection of a dead neighbor.

If an end host uses a static route to its default gateway, this creates a single point of failure if the route becomes unavailable. End hosts will not be able to detect alternate paths.

In either case, VRRP ensures that an alternate path is always available.

Definition of a Virtual Router

To back up an IP address or addresses using VRRP, a virtual router must be configured on VRRP routers on a common LAN. A VRRP router is a physical router running VRRP. A virtual router is defined by a virtual router identifier (VRID) and a set of associated IP addresses on the LAN. (On the OmniSwitch only one IP address is assigned to an interface, but other VRRP routers may have multiple IP addresses per interface. In addition, the VRID must be unique.)

Note. A limitation of the OmniSwitch is that a single VRID may only be associated with one VLAN.

Each VRRP router may back up one or more virtual routers. The VRRP router that contains the physical interfaces to which the virtual router IP addresses are assigned is called the *IP address owner*. If it is available, the IP address owner will function as the master router. The master router assumes the responsibility of forwarding packets sent to the IP addresses associated with the virtual router and answering ARP requests for these addresses.

To minimize network traffic, only the master router sends VRRP advertisements on the LAN. The IP address assigned to the physical interface on the current master router is used as the source address in VRRP advertisements. The advertisements communicate to all VRRP routers the priority and state of the master router associated with the VRID. The advertisements are IP multicast datagrams sent to the VRRP multicast address 224.0.0.18 (as determined by the Internet Assigned Numbers Authority).

If a master router becomes unavailable, it stops sending VRRP advertisements on the LAN. The backup routers know the master is unavailable based on the following algorithm:

Master Down Interval = (3 * *Advertisement Interval*) + *Skew Time*

where *Advertisement Interval* is the time interval between VRRP advertisements, and *Skew Time* is calculated based on the VRRP router's priority value as follows:

Skew Time = (256 - *Priority*) / 256

If backup routers are configured with priority values that are close in value, there may be a timing conflict, and the first backup to take over may not be the one with the highest priority; a backup with a higher priority will then preempt the new master. The virtual router may be configured to prohibit any preemption attempts, except by the IP address owner. An IP address owner, if it is available, will always become master of any virtual router associated with its IP addresses.

Note. Duplicate IP address/MAC address messages may display when a backup takes over for a master, depending on the timing of the takeover and the configured advertisement interval. This is particularly true if more than one backup is configured.

VRRP MAC Addresses

Each virtual router has a single well-known MAC address, which is used as the source in all periodic VRRP advertisements sent by the master router, any other packets originating from the master router, and as the MAC address in ARP replies (instead of a VRRP router's physical MAC address). The address has the following format:

00-00-5E-00-01-[virtual router ID]

This mapping provides for up to 255 virtual routers on a network.

ARP Requests

Each virtual router has a single well-known MAC address, which is used as the MAC address in ARP replies instead of a VRRP router's physical MAC address. When an end host sends an ARP request to the master router's IP address, the master router responds to the ARP request using the virtual router MAC address. If a backup router takes over for the master, and an end host sends an ARP request, the backup will reply to the request using the virtual router MAC address.

Gratuitous ARP requests for the virtual router IP address or MAC address are broadcast when the OmniSwitch becomes the master router. For VRRP interfaces, gratuitous ARP requests/responses are delayed at system boot until both the IP address and the virtual router MAC address are configured.

If an interface IP address is shared by a virtual router, the routing mechanism does not send a gratuitous ARP for the IP address (since the virtual router will send a gratuitous ARP). This prevents traffic from being forwarded to the router before its routing tables are stable.

ICMP Redirects

ICMP redirects are not sent out over VRRP interfaces.

VRRP Startup Delay

When a virtual router reboots and becomes master, it may become master before its routing tables are populated. This could result in loss of connectivity to the router. To prevent the loss in connectivity, a delay is used to prevent the router from becoming master before the routing tables are stabilized; the default delay value is 45 seconds.

The startup delay may be modified to allow more or less time for the router to stabilize its routing tables.

In addition to the startup delay, the switch has an ARP delay (which not configurable).

VRRP Tracking

A virtual router's priority may be conditionally modified to prevent another router from taking over as master. Tracking policies are used to conditionally modify the priority setting whenever a VLAN, slot/port, and or IP address associated with a virtual router goes down.

A tracking policy consists of a tracking ID, the value amount used to decrease the priority value, and the VLAN ID, slot/port number, or IP address to be monitored by the policy. The policy is then associated with one or more virtual routers.

Interaction With Other Features

- IP routing—IP routing must be enabled for the VRRP configuration to take effect.
- Router Discovery Protocol (RDP)—If RDP is enabled on the switch, and VRRP is enabled, RDP will advertise VLAN IP addresses of virtual routers depending on whether there are virtual routers active on the LAN, and whether those routers are backups or masters. When there are no virtual routers active on the VLAN (either acting as master or backup), RDP will advertise all VLAN IP addresses. However, if virtual routers are active, RDP will advertise IP addresses for any master routers; RDP will not advertise IP addresses for backup routers.

  For more information about RDP, see Chapter 16, "Configuring RDP."

Configuration Overview

VRRP is part of the base software. At startup, VRRP is loaded onto the switch and is enabled. Virtual routers must first be configured and enabled as described in the sections. Since VRRP is implemented on multiple switches in the network, some VRRP parameters must be identical across switches:

- **VRRP and ACLs**
  If QoS filtering rules (Access Control Lists) are configured for Layer 3 traffic on a VRRP router, all of the VRRP routers on the LAN must be configured with the same filtering rules; otherwise the security of the network will be compromised. For more information about filtering, see Chapter 25, "Configuring ACLs."

- **Conflicting VRRP Parameters Across Switches**
  All virtual routers with the same VRID on the LAN should be configured with the same advertisement interval and IP addresses. If the virtual routers are configured differently, it may result in more than one virtual router acting as the master router. This in turn would result in duplicate IP and MAC address messages as well as multiple routers forwarding duplicate packets to the virtual router MAC address. Use the **show vrrp statistics** command to check for conflicting parameters. For information about configuring VRRP parameters, see the remaining sections of this chapter.

Basic Virtual Router Configuration

At least two virtual routers must be configured on the LAN—a master router and a backup router. The virtual router is identified by a number called the Virtual Router ID (VRID), the VLAN on which the virtual router is configured, and the IP address or addresses associated with the router. Multiple virtual routers may be configured on a single physical VRRP router.

Basic commands for setting up virtual routers include:

vrrp
vrrp ip

The next sections describe how to use these commands.

Creating a Virtual Router

To create a virtual router, enter the **vrrp** command with the desired VRID and the relevant VLAN ID. The VRID must be a unique number in the range from 1 to 255. The VLAN must already be created on the switch through the **vlan** command. For information about creating VLANs, see Chapter 5, "Configuring VLANs." For example:

```
-> vrrp 23 4
```

This command creates VRID 23 on VLAN 4.

When you create a new virtual router, the VRID ID and a VLAN ID are *required*. Optionally, you may also specify:

- ***Priority*** (in the range from 1 to 255); use the **priority** keyword with the desired value. The default is 100. Note that the IP address owner will be automatically assigned a value of 255 if you do not specify the priority. See "Configuring Virtual Router Priority" on page 18-10 for more information about how priority is used.

- ***Preempt mode***. By default, preempt mode is enabled. Use **no preempt** to turn it off, and **preempt** to turn it back on. For more information about the preempt mode, see "Setting Preemption for Virtual Routers" on page 18-11.
- ***Advertising interval*** (in seconds). Use the **interval** keyword with the desired number of seconds for the delay in sending VRRP advertisement packets. The default is 1 second. See "Configuring the Advertisement Interval" on page 18-10.

The following example creates a virtual router (with VRID 24) on VLAN 2 with a priority of 75. and VRRP messages will be sent at intervals of 2 seconds.

```
-> vrrp 24 2 priority 75 no preempt interval 2
```

Note. All virtual routers with the same VRID on the same LAN should be configured with the same advertising interval; otherwise the network may produce duplicate IP or MAC address messages.

The **vrrp** command may also be used to specify whether the virtual router is enabled or disabled (it is disabled by default). *However, the virtual router must have an IP address assigned to it before it can be enabled*. Use the **vrrp ip** command as described in the next section to specify an IP address or addresses.

For more information about the **vrrp** command syntax, see the *OmniSwitch CLI Reference Guide*.

Specifying an IP Address for a Virtual Router

An IP address must be specified before a virtual router may be enabled. To specify an IP address for a virtual router, use the **vrrp ip** command and the relevant IP address. For example:

```
-> vrrp 23 4 ip 10.10.2.3
-> vrrp 23 4 enable
```

In this example, the **vrrp ip** command specifies that virtual router 23 on VLAN 4 will be used to backup IP address 10.10.2.3. The virtual router is then enabled with the **vrrp** command.

Currently the OmniSwitch does not support multiple IP addresses on a single virtual router. If an OmniSwitch is the IP address owner for a virtual router, then that address must be assigned to the virtual router. If the OmniSwitch is configured as a backup for a VRRP router that allows more than one IP address to be assigned to a virtual router, then multiple addresses may be assigned to the virtual router.

To remove an IP address from a virtual router, use the **no** form of the **vrrp ip** command. For example:

```
-> vrrp 23 4 disable
-> vrrp 23 4 no ip 10.10.2.3
```

In this example, virtual router 23 is disabled. (A virtual router must be disabled before IP addresses may be added/removed from the router.) IP address 10.10.2.3 is then removed from the virtual router with the **no** form of the **vrrp ip** command.

Configuring the Advertisement Interval

The advertisement interval is configurable, but all virtual routers with the same VRID should be configured with the same value. Mismatched values will create network problems.

If you change the advertisement interval on the master router when VRRP is already running or if the advertisement interval is set differently for a master router and a backup router, VRRP packets may be dropped because the newly configured interval does not match the interval indicated in the packet. The backup router will then take over and send a gratuitous ARP, which includes the virtual router IP address and the virtual router MAC address. In addition to creating duplicate IP/MAC address messages, both routers will begin forwarding packets sent to the virtual router MAC address. This will result in forwarding duplicate packets.

To avoid duplicate addresses and packets, make sure the advertisement interval is configured the same on both the master and the backup router.

For more information about VRRP and ARP requests, see "ARP Requests" on page 18-6.

To configure the advertisement interval, use the **vrrp** command with the **interval** keyword. For example:

```
-> vrrp 23 4 disable
-> vrrp 23 4 interval 5
```

In this example, virtual router 23 is disabled. (If you are modifying an existing virtual router, the virtual router must be disabled before it may be modified.) The **vrrp** command is then used to set the advertising interval for virtual router 23 to 5 seconds.

Configuring Virtual Router Priority

VRRP functions with one master virtual router and at least one backup virtual router. A priority value determines how backup routers will be selected to take over for the master router if it becomes unavailable.

Priority values range from 1 to 254. A value of 255 indicates that the virtual router owns the IP address; that is, the router contains the real physical interface to which the IP address is assigned. The default priority value is 100; however the switch sets this value to 255 if it detects that this router is the IP address owner. The value cannot be set to 255 if the router is not the IP address owner. The IP address owner will always be the master router if it is available.

If more than one backup router is configured, their priority values may be configured with different values, so that the backup with the higher value will take over for the master. The priority parameter may be used to control the order in which backup routers will take over for the master. If priority values are the same, any backup will take over for master.

Note that the switch sets the priority value to zero in the last VRRP advertisement packet before a master router is disabled (see "Enabling/Disabling a Virtual Router" on page 18-11).

Also, if a router is the IP address owner and the priority value is not set to 255, the switch will set its priority to 255 when the router is enabled.

To set the priority, use the **vrrp** command with the **priority** keyword and the desired value. For example:

```
-> vrrp 23 4 disable
-> vrrp 23 4 priority 50
```

In the above example, virtual router 23 is disabled. (If you are modifying an existing virtual router, the virtual router must be disabled before it may be modified.) The virtual router priority is then set to 50. The priority value is relative to the priority value configured for other virtual routers backing up the same IP address. Since the default priority is 100, setting the value to 50 would typically provide a router with lower priority in the VRRP network.

Setting Preemption for Virtual Routers

When a master virtual router becomes unavailable (goes down for whatever reason), a backup router will take over. There may be more than one backup router, and if the backup routers have similar priority values, the backup with the highest priority value may not be the one to take over for the master because of network traffic loads. If that's the case, the backup with the higher priority will then preempt the first backup router.

By default virtual routers are allowed to preempt each other; that is, if the virtual router with the highest priority will take over if the master router becomes unavailable. The preempt mode may be disabled so that any backup router that takes over when the master is unavailable will not then be preempted by a backup with a higher priority.

Note. The virtual router that owns the IP address(es) associated with the physical router always becomes the master router if is available, regardless of the preempt mode setting and the priority values of the backup routers.

To disable preemption for a virtual router, use the **vrrp** command with the **no preempt** keywords. For example:

```
-> vrrp 23 4 disable
-> vrrp 23 4 no preempt
```

In this example, virtual router 23 is disabled. (If you are modifying an existing virtual router, the virtual router must be disabled before it may be modified.) The virtual router is then configured to disable preemption. If this virtual router takes over for an unavailable router, a router with a higher priority will not be able to preempt it. For more information about priority, see "Configuring Virtual Router Priority" on page 18-10.

Enabling/Disabling a Virtual Router

Virtual routers are disabled by default. To enable a virtual router, use the **vrrp** command with the **enable** keyword. Note that at least one IP address must be configured for the virtual router through the **vrrp ip** command. For example:

```
-> vrrp 25 3 priority 150
-> vrrp ip 25 3 10.10.2.3
-> vrrp 25 3 enable
```

In this example, a virtual router is created on VLAN 3 with a VRID of 25. An IP address is then assigned to the virtual router. The virtual router is then enabled on the switch.

To disable a virtual router, use the **disable** keyword.

```
-> vrrp 25 3 disable
```

A virtual router must be disabled before it may be modified. Use the **vrrp** command to disable the virtual router first; then use the command again to modify the parameters. For example:

```
-> vrrp 25 3 disable
-> vrrp 25 3 priority 200
-> vrrp 25 3 enable
```

In this example, virtual router 25 on VLAN 3 is disabled. The virtual router is then modified to change its priority setting. (For information about configuring the priority setting, see "Configuring Virtual Router Priority" on page 18-10.) The virtual router is then re-enabled and will be active on the switch.

To delete a virtual router, use the **no** form of the **vrrp** command with the relevant VRID and VLAN ID. For example:

```
-> no vrrp 25 3
```

Virtual router 25 on VLAN 3 is deleted from the configuration. (The virtual router does not have to be disabled before you delete it.)

Setting VRRP Traps

A VRRP router has the capability to generate VRRP SNMP traps for events defined in the VRRP SNMP MIB. By default traps are enabled.

In order for VRRP traps to be generated correctly, traps in general must be enabled on the switch through the SNMP CLI. See the *OmniSwitch 7700/7800/8800 Switch Management Guide* for more information about enabling SNMP traps globally.

To disable VRRP traps, use the **no** form of the **vrrp trap** command.

```
-> no vrrp trap
```

To re-enable traps, enter the **vrrp trap** command:

```
-> vrrp trap
```

Setting VRRP Startup Delay

To set a delay to prevent a router from becoming master before its routing tables are set up, use the **vrrp delay** command.

```
-> vrrp delay 75
```

The switch will now wait 75 seconds before it will be available to take over as master for another router.

Creating Tracking Policies

To create a tracking policy, use the **vrrp track** command and specify the amount to decrease a virtual router's priority and the slot/port, IP address, or IP interface name to be tracked. For example:

```
-> vrrp track 3 enable priority 50 interface Marketing
```

In this example, a tracking policy ID (3) is created and enabled for the Marketing IP interface. If this interface goes down, a virtual router associated with this track ID will have its priority decremented by 50. Note that the **enable** keyword administratively activates the tracking policy, but the policy does not take effect until it is associated with one or more virtual routers (see the next section).

Note the following:

- A virtual router must be administratively disabled before a tracking policy for the virtual router can be added.
- VRRP tracking does not override IP address ownership (the IP address owner will always have priority to become master, if it is available).

Associating a Tracking Policy With a Virtual Router

To associate a tracking policy with a virtual router, use the **vrrp track-association** command with the tracking policy ID number. In this example, virtual router 6 on VLAN 4 is disabled first so that tracking policy 3 may be associated with it:

```
-> vrrp 6 4 disable
-> vrrp 6 4 track-association 3
```

When the virtual router is re-enabled, tracking policy 3 will be used for that virtual router. If VLAN 2 goes down, VRID 6 will have its priority decremented by 50.

A VLAN tracking policy should not be associated with a virtual router on the same VLAN. For example:

```
-> vrrp 5 2 track-association 3
```

This configuration is allowed but will not really have an effect. If VLAN 2 goes down, this virtual router goes down as well and the tracking policy is not applied.

Note. A master and a backup virtual router should not be tracking the same IP address; otherwise, when the IP address becomes unreachable, both virtual routers will have their priorities decremented, and the backup may temporarily take over if the master discovers that the IP address is unreachable before the backup.

Typically you should not configure the same IP address tracking policies on physical VRRP routers that back up each other; otherwise, the priority will be decremented for both master and backup when the entity being tracked goes down.

Verifying the VRRP Configuration

A summary of the **show** commands used for verifying the VRRP configuration is given here:

show vrrp	Displays the virtual router configuration for all virtual routers or for a particular virtual router.
show vrrp statistics	Displays statistics about VRRP packets for all virtual routers configured on the switch or for a particular virtual router.
show vrrp track	Displays information about tracking policies on the switch.
show vrrp track-association	Displays the tracking policies associated with virtual routers.

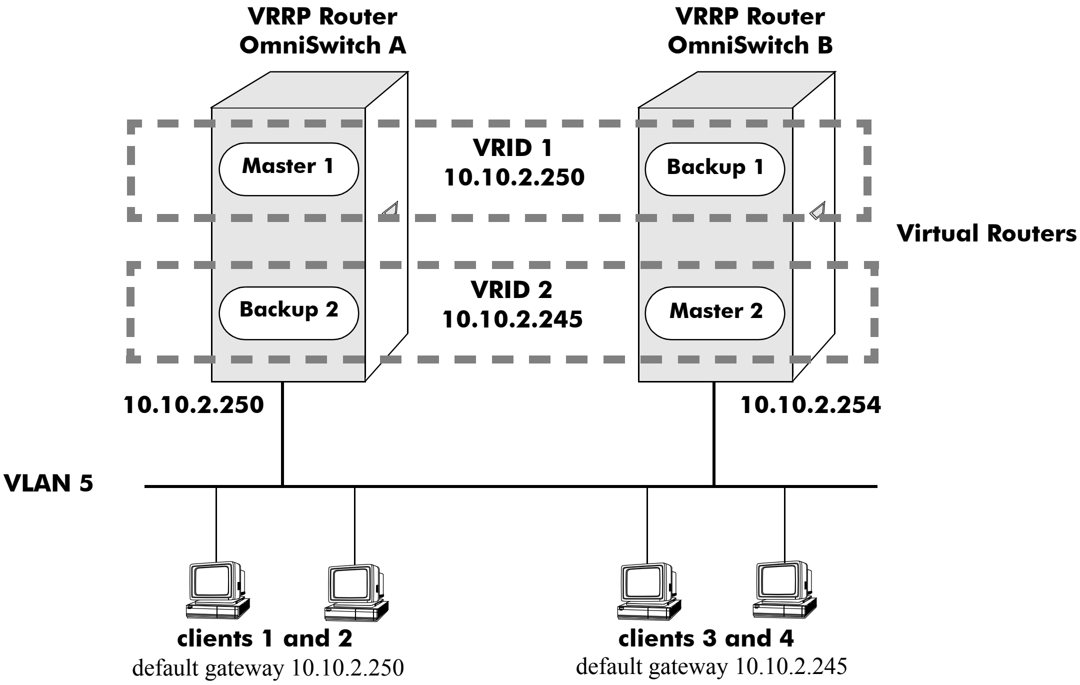
For more information about the displays that result from these commands, see the *OmniSwitch CLI Reference Guide*.

VRRP Application Examples

The following examples show how VRRP may be used to provide load balancing for outgoing traffic or to prevent loss of workstation connectivity through tracking policies.

Load Balancing Example

In addition to providing redundancy, VRRP can assist in load balancing outgoing traffic. The figure below shows two virtual routers with their hosts splitting traffic between them. Half of the hosts are configured with a default route to virtual router 1's IP address (10.10.2.250), and the other half are configured with a default route to virtual router 2's IP address (10.10.2.245).



VRRP Redundancy and Load Balancing

The CLI commands used to configure this setup are as follows:

1 First, create two virtual routers for VLAN 5. (Note that VLAN 5 must already be created and available on the switch.)

```
-> vrrp 1 5
-> vrrp 2 5
```

2 Configure the IP addresses for each virtual router.

```
-> vrrp 1 5 ip 10.10.2.250
-> vrrp 2 5 ip 10.10.2.245
```

3 Enable the virtual routers.

```
-> vrrp 1 5 enable
-> vrrp 2 5 enable
```

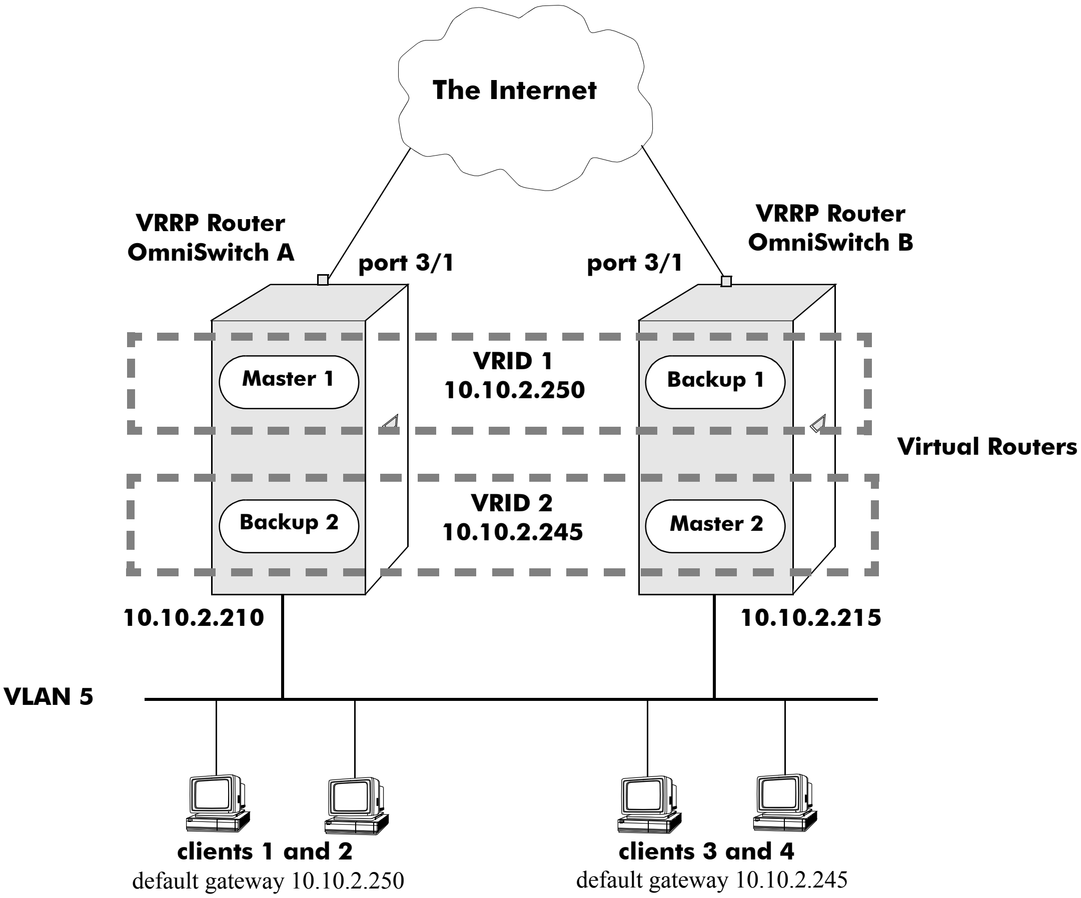
Note. The same VRRP configuration must be set up on each OmniSwitch. The VRRP router that contains, or owns, the IP address will automatically become the master for that virtual router. If the IP address is a virtual address, the virtual router with the highest priority will become the master router.

In this scenario, the master of VRID 1 will respond to ARP requests for IP address A using the virtual router MAC address for VRID 1 (00:00:5E:00:01:01). OmniSwitch 1 is the master for VRID 1 since it contains the physical interface to which 10.10.2.3 is assigned. If OmniSwitch A should become unavailable, OmniSwitch B will become master for VRID 1.

In the same way, the master of VRID 2 will respond to ARP requests for IP address B using the virtual router MAC address for VRID 2 (00:00:5E:00:01:02). OmniSwitch B is the master for VRID 2 since it contains the physical interface to which 10.10.2.245 is assigned. If OmniSwitch B should become unavailable, OmniSwitch A will become master for 10.10.2.245. This configuration provides uninterrupted service for the end hosts.

VRRP Tracking Example

The figure below shows two VRRP routers with two virtual routers backing up one IP address on each VRRP router respectively. Virtual router 1 serves as the default gateway on OmniSwitch A for clients 1 and 2 through IP address 10.10.2.250. For example, if the port that provides access to the Internet on OmniSwitch A fails, virtual router 1 will continue to be the default router for clients 1 and 2 but clients 1 and 2 will not be able to access the Internet.



VRRP Tracking Example

In this example, the master for virtual router 1 has a priority of 100 and the backup for virtual router 1 has a priority of 75. The virtual router configuration for VRID 1 on VRRP router A is as follows:

```
-> vrrp 1 5 priority 100
```

The virtual router configuration for VRID 1 on VRRP router B is as follows:

```
-> vrrp 1 5 priority 75 preempt
```

To ensure workstation clients 1 and 2 have connectivity to the internet, configure a tracking policy on VRRP router A to monitor port 3/1 and associate the policy with VRID 1.

```
-> vrrp track 1 enable priority 50 port 3/1
-> vrrp 1 5 track-association 1
```

If port 3/1 on VRRP router A goes down, the master for virtual router A is still functioning but workstation clients 1 and 2 will not be able to get to the Internet. With this tracking policy enabled, however, master router 1's priority will be temporarily decremented to 50, allowing backup 1 to take over and provide connectivity for those workstations. When port 3/1 on VRRP router A comes back up, master 1 will take over again.

Note. The preempt option must be enabled on virtual router 1; otherwise the original master will not be able to take over. See "Setting Preemption for Virtual Routers" on page 18-11 for more information about enabling preemption.

19 Configuring IPX

The Internet Packet Exchange (IPX) protocol, developed by Novell for NetWare, is a Layer 3 protocol used to route packets through IPX networks. (NetWare is Novell's network server operating system.)

In This Chapter

This chapter describes IPX and how to configure it through the Command Line Interface (CLI). It includes instructions for configuring IPX routing, and fine-tuning IPX using optional IPX configuration parameters (e.g., IPX packet extension, type-20 propagation). It also details IPX filtering, which is used to control the operation of the IPX RIP/SAP protocols. CLI commands are used in the configuration examples; for more details about the syntax of commands, see the *OmniSwitch CLI Reference Guide*.

This chapter provides an overview of IPX and includes information about the following procedures:

IPX Specifications

RFCs Supported	RFC 1792–TCP/IPX Connection MIB Specification RFC 1791–TCP And UDP Over IPX Networks With Fixed Path MTU RFC 1420–SNMP over IPX RFC 1298–SNMP over IPX

IPX Defaults

The following table lists the defaults for IPX configuration through the **ipx** command.

Description	Command	Default
IPX Status	**ipx routing**	enabled
Type-20 Packet Forwarding	**ipx type-20-propagation**	disabled
Extended RIP/SAP Packets	**ipx packet-extension**	disabled
RIP/SAP Timers	**ipx timers**	60 (seconds)

Quick Steps for Configuring IPX Routing

When IPX is enabled, devices connected to ports on the same VLAN are able to communicate. However, to route packets to a device on a different VLAN, you must create an IPX router port on each VLAN. The following steps show you how to enable IPX routing between VLANs “from scratch”. If active VLANs have already been created on the switch, go to Step 5.

1 Create VLAN 1 with a description (e.g., VLAN 1) using the **vlan** command. For example:

```
-> vlan 1 name “VLAN 1”
```

2 Create VLAN 2 with a description (e.g., VLAN 2) using the **vlan** command. For example:

```
-> vlan 2 name “VLAN 2”
```

3 Assign an active port to VLAN 1 using the **vlan port default** command. For example, the following command assigns port 1 on slot 1 to VLAN 1:

```
-> vlan 1 port default 1/1
```

4 Assign an active port to VLAN 2 using the **vlan port default** command. For example, the following command assigns port 2 on slot 1 to VLAN 2:

```
-> vlan 2 port default 1/2
```

5 Create an IPX router port on VLAN 1 using the **vlan router ipx** command. For example:

```
-> vlan 1 router ipx 00000111
```

6 Create an IPX router port on VLAN 2 using the **vlan router ipx** command. For example:

```
-> vlan 2 router ipx 00000222
```

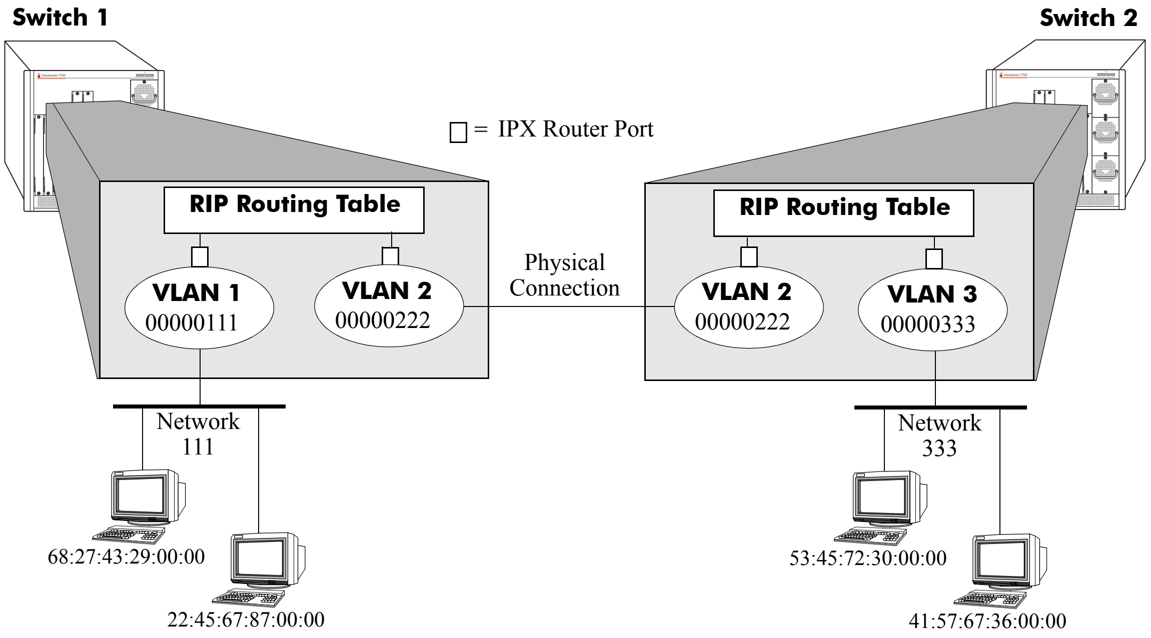
Note. For more information on VLANs and router ports, see "Configuring VLANs" on page 5-1.

IPX Overview

IPX specifies a connectionless datagram similar to the IP packet of TCP/IP networks. An IPX network address consists of two parts: a network number and a node number. The IPX network number is assigned by the network administrator. The node number is the Media Access Control (MAC) address for a network interface in the end node.

IPX exchanges information using its own version of RIP, which sends updates every 60 seconds. NetWare also supports SAP to allow network resources, including file and print servers, to advertise their network addresses and the services they provide. The user can also define routes. These routes, called static routes, have higher priority than routes learned through RIP.

When IPX is enabled, devices connected to ports on the same VLAN are able to communicate. However, to route packets between VLANs, you must create an IPX router port on each VLAN. In the illustration below, a router port has been configured on each VLAN. Therefore, workstations connected to ports on VLAN 1 on Switch 1 can communicate with VLAN 2; and workstations connected to ports on VLAN 3 on Switch 2 can communicate with VLAN 2. Also, ports from both switches have been assigned to VLAN 2, and a physical connection has been made between the switches. Therefore, workstations connected to VLAN 1 on Switch 1 can communicate with workstations connected to VLAN 3 on Switch 2.



IPX Routing

In IPX routing, the switch builds routing tables to keep track of optimal destinations for traffic it receives that is destined for remote IPX networks. The switch sends and receives routing messages, or advertisements, to/from other switches in the network. When the switch receives an IPX packet, it looks up the destination network number in its routing table. If the network is directly connected to the switch, the switch also checks the destination node address.

IPX is associated with additional protocols built into the switch software. The switch supports the following IPX protocols:

- **IPX** RIP—Layer 3 protocol used by NetWare routers to exchange IPX routing information. IPX RIP functions similarly to IP RIP. IPX RIP uses two metrics to calculate the best route: hop count and ticks. An IPX router periodically transmits packets containing the information currently in its own routing table to neighboring IPX RIP routers to advertise the best route to an IPX destination.
- SAP—Layer 3 protocol used by NetWare routers to exchange IPX routing information. SAP is similar in concept to IPX RIP. Just as RIP enables NetWare routers to exchange information about routes, SAP enables NetWare devices to exchange information about available network services. NetWare workstations use SAP to obtain the network addresses of NetWare servers. IPX routers use SAP to gather service information and then share it with other IPX routers.
- Sequenced Packet Exchange (SPX)—Transport-layer protocol that provides a reliable end-to-end communications link by managing packet sequencing and delivery. SPX does not play a direct role in IPX routing; it simply guarantees the delivery of routed packets.

IPX Routing

When IPX is enabled, devices connected to ports on the same VLAN are able to communicate. However, to route packets to a device on a different VLAN, you must create an IPX router port on each VLAN.

Enabling IPX Routing

IPX is enabled by default. If necessary, use the **ipx routing** command to enable IPX. Use the **no ipx routing** command to disable IPX. Use the **show ipx interface** command to display IPX router status and configuration parameters.

Creating an IPX Router Port

You must configure an IPX router port on a VLAN for devices on that VLAN to communicate with devices on other VLANs. You can only create one IPX router port per VLAN. VLAN router ports are not active until at least one active physical port is assigned to the VLAN.

If the switch is currently in single mac router mode, up to 256 router ports are supported (including IP and IPX). If the switch is in multiple mac router mode, up to 64 router ports are supported (including IP and IPX). You can configure an IP and IPX router port on the same VLAN. Both types of router ports will share the same MAC address for that VLAN.

Use the **vlan router ipx** command to configure an IPX router port. For example, to create an IPX router port on VLAN 1 with an IPX address of 1000590C, you would enter:

```
-> vlan 1 router ipx 1000590C
```

Note. If fewer than eight hex digits are entered for an IPX network number, the entry is automatically prefixed with zeros to equal eight digits.

Use the **no vlan router ipx** command to remove an IPX router port from the VLAN. For example, to remove an IPX router port on VLAN 1 with an IPX address of 1000590C, you would enter:

```
-> no vlan 1 router ipx 1000590C
```

Use the **show ipx interface** command to display current IPX interface information.

Note. Router port IPX addresses must be unique. You cannot have two router ports with the same IPX address.

For more information on VLANs, see Chapter 5, “Configuring VLANs.”

IPX Router Port Configuration Options

When you create an IPX router port using the **vlan router ipx** command, RIP routing is enabled using the default parameters listed below. However, you can use the full command to change the default parameters. Sample configurations are shown at the end of this section.

Routing Type

By default both RIP and SAP packets are processed (active). However, additional configurations can be used:

- **active.** RIP and SAP updates are processed (default)
- **rip.** RIP updates are processed (SAP is disabled)
- **inactive.** RIP and SAP updates are not processed, but the router port remains active.

Encapsulation Type

Ethernet 2 encapsulation is the default encapsulation type. However, other types can be configured:

- **e2.** Ethernet 2 encapsulation (default)
- **novell.** Novell Raw (802.3) encapsulation
- **llc.** LLC (802.2) encapsulation
- **snap.** SNAP encapsulation.

Delay

To configure the IPX delay, enter the syntax **timeticks** and specify the number of ticks for IPX delay time. A tick is approximately 1/18th of a second. The valid range is 0–65535. The default is 0.

For example, to configure IPX router port 1000590C on VLAN 1 to process only RIP packets with a delay of 10 you would enter:

```
-> vlan 1 router ipx 1000590C rip timeticks 10
```

For more information on optional command syntax see Chapter 1, "VLAN Management Commands" in the *OmniSwitch CLI Reference Guide*. For more information on VLANs and configuring router ports, see Chapter 5, "Configuring VLANs."

Creating/Deleting a Default Route

A default IPX route can be configured for packets destined for networks unknown to the switch. If RIP is disabled, and a default IPX route is configured, packets can still be forwarded to a switch that knows where to send them.

Use the **ipx default-route** command to configure a default route for the switch. Enter the command, then enter the IPX network number of the first hop used to reach the default route. For example, to configure a default route using IPX network 222 for the first hop you would enter:

```
-> ipx default-route 222
```

The IPX network number is required. You can also enter the VLAN number of the first hop. For example, to configure a default route using VLAN 1 on the 222 network you would enter:

```
-> ipx default-route 1 222
```

The network node is only required if the default network is directly connected to the switch. For example, to create a default route to network 222 (which is directly attached to the switch) you would enter:

```
-> ipx default-route 222 00:20:da:99:88:77
```

Use the **no ipx default-route** command to delete a default route. For example, to delete a default route using the 222 network as a first hop you would enter:

```
-> no ipx default-route 222
```

Use the **show ipx default-route** command to display IPX default routes.

Creating/Deleting Static Routes

A static route enables you to send traffic to a switch other than those learned through routing protocols. Static routes are are user-defined and carry a higher priority than routes created by dynamic routing protocols. That is, if two routes have the same metric value, the static route has the higher priority. Static routes allow you to define, or customize, an explicit path to an IP network segment, which is then added to the IP forwarding table. Static routes can be created between VLANs to enable devices on these VLANs to communicate.

Use the **ipx route** command to configure a static route for the switch. Enter the IPX network number of the route's final destination, then enter the IPX network and node numbers used to reach the first hop of the route. You can also enter the optional parameters of hop count (number of hops to the destination network) and delay. The delay is the time, in ticks, to reach the route's destination. One tick is equivalent to 1/18 of a second (approximately 55ms).

For example to create a static route to network 222 with a first hop network of 0000590C, node 00:20:da:99:88:77 you would enter:

```
-> ipx route 222 590C 00:20:da:99:88:77
```

Static routes do not age out of the routing tables; however, they can be deleted. Use the **no ipx route** command to delete a static route. To delete a static route, you only need to enter the network number of the destination node. For example, to delete a static route to network 222 you would enter:

```
-> no ipx route 222
```

Use the **show ipx route** command to display IPX routes.

Configuring Type-20 Packet Forwarding

Type 20 is an IPX packet type that refers to any propagated packet. Novell has defined the use of these packets to support certain protocol implementations, such as NetBIOS. Because these packets are broadcast and propagated across networks, the addresses of those networks (up to 8) are stored in the packet's data area. If Type 20 packet forwarding is enabled, the switch receives and propagates Type 20 packets through all its interfaces. If Type 20 packet forwarding is disabled, the switch discards, rather than propagates, any Type 20 packet it receives. Type 20 packet forwarding is disabled by default. This is because these packets can cause problems in highly redundant IPX networks by creating what appears to be a broadcast storm. This problem is aggravated whenever misconfigured PCs are added to a network.

Use the **ipx type-20-propagation** command to enable or disable Type 20 packet forwarding on the switch. For example:

```
-> ipx type-20-propagation enable
```

You can also enable or disable Type 20 packet forwarding on a specific VLAN using the optional VLAN parameter. For example, to enable Type 20 packet forwarding only on VLAN 1 you would enter:

```
-> ipx type-20-propagation 1 enable
```

Use the **show ipx type-20-propagation** command to display Type 20 packet forwarding status for the switch.

Configuring Extended RIP and SAP Packets

Larger RIP and SAP packets can be transmitted to reduce network congestion. Other switches and routers in the network must support larger packet sizes if this feature is configured on the switch. RIP packets can contain up to 68 network entries. SAP packets can contain up to 8 network entries. Extended RIP and SAP packets are disabled by default.

Use the **ipx packet-extension** command to enable or disable extended RIP/SAP packets on the switch. For example:

```
-> ipx packet-extension enable
```

You can also enable or disable extended RIP/SAP packets on a specific VLAN using the optional VLAN parameter. For example, to enable extended RIP/SAP packets only on VLAN 1 you would enter:

```
-> ipx packet-extension 1 enable
```

Use the **show ipx packet-extension** command to display extended RIP/SAP packet status for the switch.

Configuring RIP and SAP Timers

By default, RIP and SAP packets are broadcast every 60 seconds, even if no change has occurred anywhere in a route or service. This default may be modified to alleviate network congestion or facilitate the discovery of network resources.

Use the **ipx timers** command to set the RIP/SAP broadcast time for the switch. You must set both the RIP and SAP timer values. For example, to set a RIP timer value of 120 and a SAP timer value of 180 you would enter:

```
-> ipx timers 120 180
```

Use the **no ipx timers** command to return the timer values to the default of 60.

You can set the RIP/SAP timers on a specific VLAN using the optional VLAN parameter. For example, to set a RIP timer value of 120 and a SAP timer value of 180 on VLAN 1 you would enter:

```
-> ipx timers 1 120 180
```

Use the **show ipx timers** command to display the current RIP/SAP timer values.

Using the PING Command

The ping command is used to test the reachability of certain types of IPX nodes. The software supports two different types of IPX pings:

- Novell—Used to test the reachability of NetWare servers currently running the NetWare Loadable Module called IPXRTR.NLM. This type *cannot* be used to reach NetWare workstations running IPXODI. Novell uses a unique type of ping for this purpose (implemented by their IPXPNG.EXE program). This type of ping is not currently supported by the switch software. Other vendors' switches may respond to this type of ping.
- Alcatel—Used to test the reachability of Alcatel switches on which IPX routing is enabled.

Network devices that do not recognize the specific type of IPX ping request sent from the switch will not respond at all. This lack of a response does not necessarily mean that a specific network device is inactive or missing. Therefore, you might want to try using both types before concluding that the network device is "unreachable."

Use the **ping ipx** command to ping an IPX Node. Enter the command, followed by the network and network node number of the device you want to ping. The packet will use the default parameters for count (5), size (64), timeout (1), and type (novell). For example, to ping an IPX device (node 00:20:da:05:16:94) on IPX network 304 you would enter:

```
-> ping ipx 304 00:20:da:05:16:94
```

When you ping a device, the device IPX address and node are *required*. Optionally, you may also specify:

- **Count.** Use the **count** keyword to set the number of packets to be transmitted.
- **Size.** Use the **size** keyword to set the size, in bytes, of the data portion of the packet sent for this ping. The valid range is 1 to 8192.
- **Timeout.** Use the **timeout** keyword to set the number of seconds the program will wait for a response before timing out.

- **Type.** Use the **type** keyword to specify the packet type you want to send (**novell** or **alcatel**). Use the **novell** packet type to test the reachability of NetWare servers running the NetWare Loadable Module (IPXRTR.NLM). This type cannot be used to reach NetWare workstations running IPXODI. You can use the **alcatel** packet type to test the reachability of Alcatel switches on which IPX routing is enabled. However, Alcatel switches will respond to either type.

For example, to send a ping with a count of 2, a size of 32 bytes, a timeout of 10 seconds, that is an **alcatel** type packet you would enter:

```
-> ping ipx 304 00:20:da:05:16:94 count 2 size 32 timeout 10 type alcatel
```

Note. If you change the default values they will only apply to the current ping. The next time you use the ping command, the default values will be used unless you again enter different values.

IPX RIP/SAP Filtering

The IPX RIP/SAP Filtering feature give you a means of controlling the operation of the IPX RIP/SAP protocols. By using IPX RIP/SAP filters, you can minimize the number of entries put in the IPX RIP Routing and SAP Bindery Tables, improve overall network performance by eliminating unnecessary traffic, and control users' access to NetWare services. For example:

- RIP Input and Output filters can be used to isolate entire network segments (and/or switches) to make the network appear differently to the different segments.

- RIP Input and Output filters can be used to reduce the amount of traffic needed to advertise routes that should not be used by a particular network segment.

- SAP Input and Output filters can be used to improve performance by limiting the amount of SAP traffic. For example, because printing is generally a local operation, there's no need to advertise print servers to remote networks. A SAP filter can be used in this case to restrict "Print Server Advertisement" SAPs.

Five types of IPX RIP/SAP filters are available:

- **RIP Input Filters.** Control which networks are allowed into the routing table when IPX RIP updates are received.

- **RIP Output Filters.** Control the list of networks included in routing updates sent by the switch. These filters control which networks the switch advertises in its IPX RIP updates.

- **SAP Input Filters.** Control the SAP updates received by the switch prior to a switch accepting information about a service. The switch will filter all incoming service advertisements received before accepting information about a service.

- **SAP Output Filters.** Control which services are included in SAP updates sent by the switch. The switch applies the SAP output filters prior to sending SAP packets.

- **GNS Output Filters.** Control which servers are included in the GNS responses sent by the switch.

All types of IPX Filters can be configured either to allow or to block traffic. The default setting for all filters is to allow traffic. Therefore, you will typically only have to define a filter to block traffic. However, defining a filter to allow certain traffic may be useful in situations where a more generic filter has been defined to block the majority of the traffic. For example, you could use a filter to allow traffic from a specific host on a network where all other traffic has been blocked. A discussion of the precedence of "allow" filters appears later in this section. Keep in mind that precedence applies only to "allow" filters, not to "block" filters.

Note. You can apply filters to all router interfaces by defining a "global" filter, or you can limit the filter to specific interfaces.

Configuring RIP Filters

IPX RIP filters allow you to minimize the number of entries put in the IPX RIP routing table. RIP input filters control which networks are allowed into the routing table when IPX RIP updates are received. RIP output filters control which networks the switch advertises in its IPX RIP updates.

Use the **ipx filter rip** command to configure a RIP input or output filter. To configure a global filter that will be applied to all traffic, enter the command, specify whether it is an input (**in**) or output (**out**) filter, then specify whether you want the filter to allow or block traffic. For example, to create a filter that will block all incoming RIP packets you would enter:

```
-> ipx filter rip in block
```

You can narrow the filter by specifying a VLAN. For example, to create a filter that will block all incoming RIP packets from VLAN 1 you would enter:

```
-> ipx filter 1 rip in block
```

You can also narrow the filter by specifying a network. You must enter the network number and the network mask. For example, to create a filter that will block incoming RIP packets from network 40 and its subnets you would enter:

```
-> ipx filter rip in block 40 mask ffffffff
```

Use the **no ipx rip filter** command to delete a RIP filter. For example, to delete a global RIP filter that was configured to block incoming RIP packets you would enter:

```
-> no ipx filter rip in block
```

Use the optional syntax to delete a filter for a specific VLAN or network. If you are deleting the filter for a specific network you can also enter the network mask. To delete a filter from all VLANs/networks, use only the basic command syntax (e.g., **no ipx filter rip in allow**).

Use the **show ipx filter** command to display all IPX filters.

Note. RIP filters work only on switches running the RIP protocol. They do not work for switches running the NLSP protocol. Use RIP filters with care because they can partition a physical network into two or more segments.

Configuring SAP Filters

IPX SAP filters allow you to minimize the number of entries put in the SAP Bindery Table. SAP input filters control the SAP updates received by the switch prior to a switch accepting information about a service. The switch will filter all incoming service advertisements received before accepting information about a service. SAP output filters control which services are included in SAP updates sent by the switch.

Use the **ipx filter sap** command to configure a SAP input or output filter. To configure a global filter that will be applied to all traffic, enter the command, specify the SAP packet type to be filtered (**all** – all SAP packets, or a specific 4-digit hex SAP type), specify whether it is an input (**in**) or output (**out**) filter, then specify whether you want the filter to allow or block traffic. For example, to block all SAP updates sent by the switch you would enter:

```
-> ipx filter sap all out block
```

You can narrow the filter by specifying a VLAN and SAP type. For example, to create a filter that will block 0004 (NetWare File Server) SAP updates from being sent to VLAN 1 you would enter:

```
-> ipx filter 1 sap 0004 out block
```

You can also narrow the filter by specifying a network. You must enter the network number and the network mask. For example, to create a filter that will block 0004 SAP updates from being sent to network 222 and its subnets you would enter:

```
-> ipx filter sap 0004 out block 222 mask fffffff
```

Use the **no ipx sap filter** command to delete a SAP filter. For example, to delete a global SAP filter that was configured to block incoming SAP packets you would enter:

```
-> no ipx filter sap in block
```

Use the optional syntax to delete a filter for a specific VLAN or network. If you are deleting the filter for a specific network you can also enter the network mask. To delete a filter from all VLANs/networks, use only the basic command syntax (e.g., **no ipx filter sap in allow**).

Use the **show ipx filter** command to display all IPX filters.

Configuring GNS Filters

GNS output filters control which servers are included in the GNS responses sent by the switch. GNS supports output filters only.

Use the **ipx filter gns** command to configure a GNS filter. To configure a global filter that will be applied to all traffic, enter the command, specify the GNS packet type to be filtered (**all** – all GNS packets, or a specific 4-digit hex GNS type), specify whether it is an input (**in**) or output (**out**) filter, then specify whether you want the filter to allow or block traffic. For example to block all GNS updates you would enter:

```
-> ipx filter gns all out block
```

You can narrow the filter by specifying a VLAN. For example to block all GNS updates sent to VLAN 1 you would enter:

```
-> ipx filter 1 gns all out block
```

You can also narrow the filter by specifying a network. You must enter the network number and the network mask. For example, to create a filter that will block updates sent to network 222 and its subnets you would enter:

```
-> ipx filter gns all out block 222 mask ffffffff
```

Use the **no ipx gns filter** command to delete a GNS filter. For example, to delete a global GNS filter that was configured to block all GNS updates you would enter:

```
-> no ipx filter gns all out block
```

Use the **show ipx filter** command to display all IPX filters.

IPX RIP/SAP Filter Precedence

Whenever you use multiple “allow” filters you must first define a filter to block all RIPs or SAPs. Then, all of the subsequent “allow” filters of the same type must be at least as specific in all areas for the filters to work. Note that filtering precedence is related only to “allow” filters. Multiple “block” filters can be defined with varying specificity in each of the areas of the filter.

For example, consider a switch that knows of multiple Type 0004 SAPs on various networks, including a network with an address of “40.” The switch also knows of various types of SAPs on Network 40. For this example, you want to block all SAP updates coming from Network 40, but you want to allow all Type 0004 SAPs, including the ones that come from Network 40. To meet these objectives, you would configure the following filters:

Filter 1

ipx filter sap all in block 40 mask fffffff

This filter will block all SAP Type updates on all nodes of network 40.

Filter 2

ipx filter sap 0004 in allow 40 mask fffffff

This filter will allow only SAP Type 0004 updates on all nodes of network 40. It is more specific than the block filter so only SAP Type 0004 updates will be allowed.

The filters shown below will *not* work for our example because in Filter 2 the type of service is *less* specific than the type defined in Filter 1. All Type 0004 SAPs will be blocked by the filter.

Filter 1

ipx filter sap 0004 in block 40 mask fffffff

This filter will block only SAP Type 0004 updates on all nodes of network 40.

Filter 2

ipx filter sap all in allow 40 mask fffffff

This filter will allow all SAP Types on all nodes of network 40. It is less specific than the block filter so all SAP updates will be allowed.

Flushing the IPX RIP/SAP Tables

When you flush the RIP/SAP table(s) only routes learned by RIP and SAP are deleted; static routes are not removed. The RIP Table and SAP Bindery Tables can contain a maximum of 2,000 entries each.

Use the **clear ipx route** command to flush the IPX RIP and/or SAP Bindery Tables. Enter the command, followed by the table that you want to clear (rip, sap, or all). For example to clear all dynamic entries from both the RIP and SAP tables you would enter:

```
-> clear ipx route all
```

Use the **show ipx route** command to display the IPX RIP Routing Table.

Verifying the IPX Configuration

A summary of the show commands used for verifying the IPX configuration is given here:

show ipx interface	Displays current IPX interface configuration information.
show ipx route	Displays IPX routing table information.
show ipx filter	Displays currently configured IPX RIP, SAP, and GNS filters.
show ipx type-20-propagation	Displays the current status of Type 20 packet forwarding.
show ipx packet-extension	Displays the current status of the extended RIP/SAP packet feature.
show ipx timers	Displays the current RIP and SAP timer values.

For more information about the displays that result from these commands, see the *OmniSwitch CLI Reference Guide*.

20 Managing Authentication Servers

This chapter describes authentication servers and how they are used with the switch. The types of servers described include Remote Authentication Dial-In User Service (RADIUS), Lightweight Directory Access Protocol (LDAP), and SecurID's ACE/Server.

In This Chapter

The chapter includes some information about attributes that must be configured on the servers, but it primarily addresses configuring the switch through the Command Line Interface (CLI) to communicate with the servers to retrieve authentication information about users.

Configuration procedures described include:

- **Configuring an ACE/Server**. This procedure is described in "ACE/Server" on page 20-8.
- **Configuring a RADIUS Server**. This procedure is described in "RADIUS Servers" on page 20-9.
- **Configuring an LDAP Server**. This procedure is described in "LDAP Servers" on page 20-15.

For information about using servers for authenticating users to manage the switch, see the "Switch Security" chapter in the *OmniSwitch 7700/7800/8800 Switch Management Guide*.

For information about using servers to retrieve authentication information for Layer 2 Authentication users (authenticated VLANs), see Chapter 21, "Configuring Authenticated VLANs."

Authentication Server Specifications

RADIUS RFCs Supported	RFC 2865–Remote Authentication Dial In User Service (RADIUS) RFC 2866–RADIUS Accounting RFC 2867–RADIUS Accounting Modifications for Tunnel Protocol Support RFC 2868–RADIUS Attributes for Tunnel Protocol Support RFC 2809–Implementation of L2TP Compulsory Tunneling via RADIUS RFC 2869–RADIUS Extensions RFC 2548–Microsoft Vendor-specific RADIUS Attributes RFC 2882–Network Access Servers Requirements: Extended RADIUS Practices
LDAP RFCs Supported	RFC 1789–Connectionless Lightweight X.5000 Directory Access Protocol RFC 2247–Using Domains in LDAP/X.500 Distinguished Names RFC 2251–Lightweight Directory Access Protocol (v3) RFC 2252–Lightweight Directory Access Protocol (v3): Attribute Syntax Definitions RFC 2253–Lightweight Directory Access Protocol (v3): UTF-8 String Representation of Distinguished Names RFC 2254–The String Representation of LDAP Search Filters RFC 2256–A Summary of the X.500(96) User Schema for Use with LDAPv3
Other RFCs	RFC 2574–User-based Security Model (USM) for version 3 of the Simple Network Management Protocol (SNMPv3) RFC 2924–Accounting Attributes and Record Formats RFC 2975–Introduction to Accounting Management RFC 2989–Criteria for Evaluating AAA Protocols for Network Access
Maximum number of authentication servers in single authority mode	4 (not including any backup servers)
Maximum number of authentication servers in multiple authority mode	4 per VLAN (not including any backup servers)
Maximum number of servers per Authenticated Switch Access type	4 (not including any backup servers)
CLI Command Prefix Recognition	The **aaa radius-server** and **aaa ldap-server** commands support prefix recognition. See the “Using the CLI” chapter in the *OmniSwitch 7700/7800/8800 Switch Management Guide* for more information.

Server Defaults

The defaults for authentication server configuration on the switch are listed in the tables in the next sections.

RADIUS Authentication Servers

Defaults for the **aaa radius-server** command are as follows:

Description	Keyword	Default
Number of retries on the server before the switch tries a backup server	**retransmit**	3
Timeout for server replies to authentication requests	**timeout**	2
UDP destination port for authentication	**auth-port**	1645*
UDP destination port for accounting	**acct-port**	1646*

* The port defaults are based on the older RADIUS standards; some servers are set up with port numbers based on the newer standards (ports 1812 and 1813 respectively).

LDAP Authentication Servers

Defaults for the **aaa ldap-server** command are as follows:

Description	Keyword	Default
The port number for the server	**port**	389 (SSL disabled) 636 (SSL enabled)
Number of retries on the server before the switch tries a backup server	**retransmit**	3
Timeout for server replies to authentication requests	**timeout**	2
Whether a Secure Socket Layer is configured for the server	**ssl | no ssl**	no ssl

Quick Steps For Configuring Authentication Servers

1 For RADIUS or LDAP servers, configure user attribute information on the servers. See "RADIUS Servers" on page 20-9 and "LDAP Servers" on page 20-15.

2 Use the **aaa radius-server** and/or the **aaa ldap-server** command to configure the authentication server(s). For example:

```
-> aaa radius-server rad1 host 10.10.2.1 10.10.3.5 key amadeus
-> aaa ldap-server ldap2 host 10.10.3.4 dn cn=manager password tpub base c=us
```

Note. (*Optional*) Verify the server configuration by entering the **show aaa server** command. For example:

```
-> show aaa server
Server name = rad1
  Server type            = RADIUS,
  IP Address 1           = 10.10.2.1,
  IP Address 2           = 10.10.3.5
  Retry number           = 3,
  Timeout (in sec)       = 2,
  Authentication port    = 1645,
  Accounting port        = 1646
Server name = ldap2
  Server type            = LDAP,
  IP Address 1           = 10.10.3.4,
  Port                   = 389,
  Domain name            = cn=manager,
  Search base            = c=us,
  Retry number           = 3,
  Timeout (in sec)       = 2,
```

See the *OmniSwitch CLI Reference Guide* for information about the fields in this display.

3 If you are using ACE/Server, there is no required switch configuration; however, you must FTP the **sdconf.rec** file from the server to the switch's **/network** directory.

4 Configure authentication on the switch. This step is described in other chapters. For a quick overview of using the configured authentication servers with Authenticated VLANs, see "AVLAN Configuration Overview" on page 21-4. For a quick overview of using the configured authentication servers with Authenticated Switch Access, see the *OmniSwitch 7700/7800/8800 Switch Management Guide*.

Server Overview

Authentication servers are sometimes refered to as AAA servers (authentication, authorization, and accounting). These servers are used for storing information about users who want to manage the switch (Authenticated Switch Access) and users who need access to a particular VLAN or VLANs (Authenticated VLANs or Port-Based Network Access Control).

RADIUS or LDAP servers may be used for Authenticated Switch Access and/or Authenticated VLANs. Another type of server, SecurID's ACE/Server, may be used for authenticated switch access only; the ACE/Server is an authentication-only server (no authorization or accounting). Only RADIUS servers are supported for 802.1X Port-Based Network Access Control.

The following table describes how each type of server may be used with the switch:

Server Type	Authenticated Switch Access	Authenticated VLANs	802.1X Port-Based Network Access Control
ACE/Server	yes (except SNMP)	no	no
RADIUS	yes (except SNMP)	yes	yes
LDAP	yes (including SNMP)	yes	no

Backup Authentication Servers

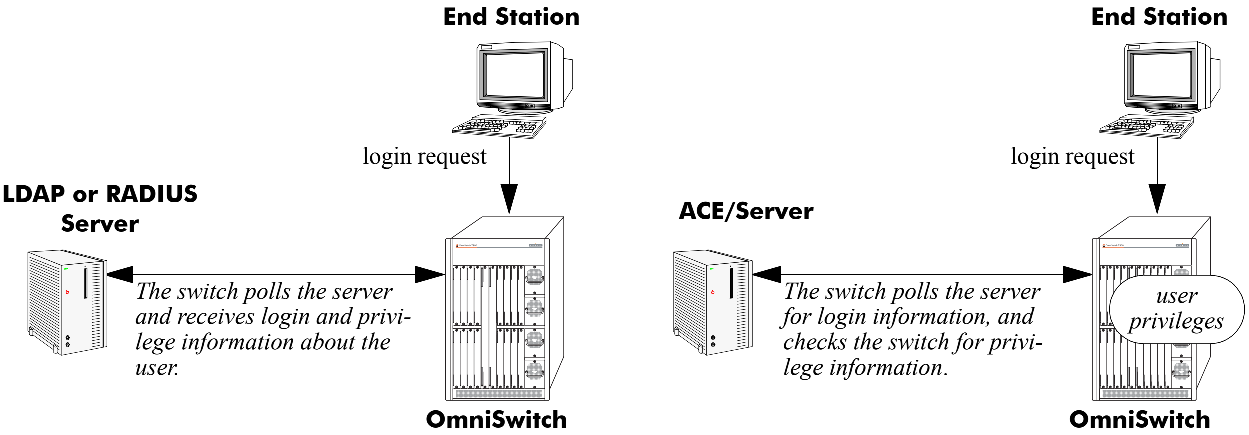
Each RADIUS and LDAP server may have one backup host (of the same type) configured through the **aaa radius-server** and **aaa ldap-server** commands respectively. In addition, each authentication method (Authenticated Switch Access, Authenticated VLANs, or 802.1X) may specify a list of backup authentication servers that includes servers of different types (if supported on the feature).

The switch uses the first available authentication server to attempt to authenticate users. If user information is not found on the first available server, the authentication attempts fails.

Authenticated Switch Access

When RADIUS and/or LDAP servers are set up for Authenticated Switch Access, the switch polls the server for user login information. The switch also polls the server for privilege information (authorization) if it has been configured on the server; otherwise, the local user database is polled for the privileges.

A RADIUS server supporting the challenge and response mechanism as defined in RADIUS RFC 2865 may access an ACE/Server for authentication purposes. The ACE/Server is then used for user authentication, and the RADIUS server is used for user authorization.

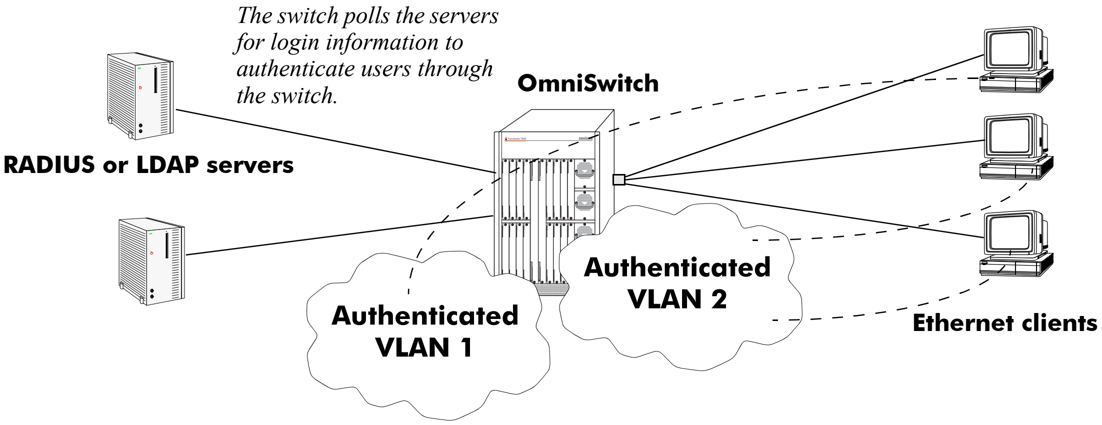

End Station
login request
LDAP or RADIUS Server
The switch polls the server and receives login and privilege information about the user.
OmniSwitch
End Station
login request
ACE/Server
The switch polls the server for login information, and checks the switch for privilege information.
user privileges
OmniSwitch

Servers Used for Authenticated Switch Access

Authenticated VLANs

For authenticated VLANs, authentication servers contain a database of user names and passwords, challenges/responses, and other authentication criteria such as time-of-day access. The Authenticated VLAN attribute is required on servers set up in multiple authority mode.

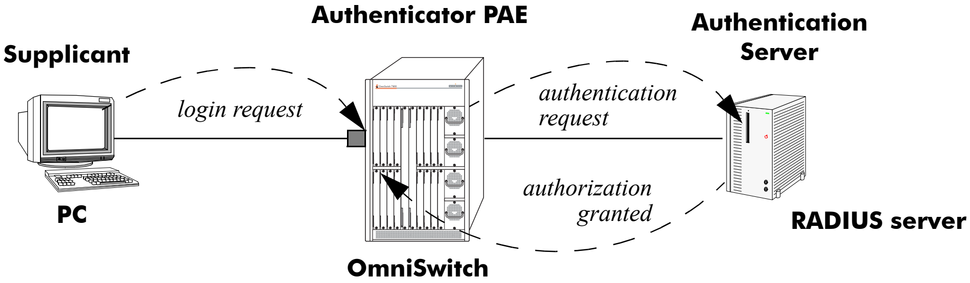
Servers may be configured using one of two different modes, single authority mode or multiple authority mode. The mode specifies how the servers are set up for authentication: single authority mode uses a single list (an authentication server and any backups) to poll with authentication requests. Multiple authority mode uses multiple lists, one list for each authenticated VLAN. For more information about authority modes and Authenticated VLANs, see Chapter 21, “Configuring Authenticated VLANs.”


The switch polls the servers for login information to authenticate users through the switch.
OmniSwitch
RADIUS or LDAP servers
Authenticated VLAN 1
Authenticated VLAN 2
Ethernet clients

Servers Used for Authenticated VLANs

Port-Based Network Access Control (802.1X)

For devices authenticating on an 802.1X port on the switch, only RADIUS authentication servers are supported. The RADIUS server contains a database of user names and passwords, and may also contain challenges/responses and other authentication criteria.



Basic 802.1X Components

For more information about configuring 802.1X ports on the switch, see Chapter 22, "Configuring 802.1X."

ACE/Server

An external ACE/Server may be used for authenticated switch access. It cannot be used for Layer 2 authentication or for policy management. Attributes are not supported on ACE/Servers. These values must be configured on the switch through the **user** commands. See the “Switch Security” chapter of the *OmniSwitch 7700/7800/8800 Switch Management Guide* for more information about setting up the local user database.

Since an ACE/Server does not store or send user privilege information to the switch, user privileges for Secur/ID logins are determined by the switch. When a user attempts to log into the switch, the user ID and password is sent to the ACE/Server. The server determines whether the login is valid. If the login is valid, the user privileges must be determined. The switch checks its user database for the user’s privileges. If the user is not in the database, the switch uses the default privilege, which is determined by the default user account. For information about the default user account, see the “Switch Security” chapter of the *OmniSwitch 7700/7800/8800 Switch Management Guide.*

There are no server-specific parameters that must be configured for the switch to communicate with an attached ACE/Server; however, you must FTP the **sdconf.rec** file from the server to the switch’s **/network** directory. This file is required so that the switch will know the IP address of the ACE/Server. For information about loading files onto the switch, see the *OmniSwitch 7700/7800/8800 Switch Management Guide*.

The ACE client in the switch is version 4.1; it does not support the replicating and locking feature of ACE 5.0, but it may be used with an ACE 5.0 server if a legacy configuration file is loaded on the server. The legacy configuration must specify authentication to two specific servers (master and slave). See the RSA Security ACE/Server documentation for more information.

To display information about any servers configured for authentication, use the **show aaa server** command. For more information about the output for this command, see the *OmniSwitch CLI Reference Guide*.

Also, you may need to clear the ACE/Server secret occasionally because of misconfiguration or required changes in configuration. Clearing the secret is described in the next section.

Clearing an ACE/Server Secret

The ACE/Server generates “secrets” that it sends to clients for authentication. While you cannot configure the secret on the switch, you can clear it. The secret may need to be cleared because the server and the switch get out of synch. See the RSA Security ACE/Server documentation for more information about the server secret.

To clear the secret on the switch, enter the following command:

```
-> aaa ace-server clear
```

When you clear the secret on the switch, the secret must also be cleared on the ACE/Server as described by the RSA Security ACE/Server documentation.

RADIUS Servers

RADIUS is a standard authentication and accounting protocol defined in RFC 2865 and RFC 2866. A built-in RADIUS client is available in the switch. A RADIUS server that supports Vendor Specific Attributes (VSAs) is required. The Alcatel attributes may include VLAN information, time-of-day, or slot/port restrictions.

RADIUS Server Attributes

RADIUS servers and RADIUS accounting servers are configured with particular attributes defined in RFC 2138 and RFC 2139, respectively. These attributes carry specific authentication, authorization, and configuration details about RADIUS requests to and replies from the server. This section describes the attributes and how to configure them on the server.

Standard Attributes

The following tables list RADIUS server attributes 1–39 and 60–63, their descriptions, and whether the Alcatel RADIUS client in the switch supports them. Attribute 26 is for vendor-specific information and is discussed in "Vendor-Specific Attributes for RADIUS" on page 20-11. Attributes 40–59 are used for RADIUS accounting servers and are listed in "RADIUS Accounting Server Attributes" on page 20-13.

Num.	Standard Attribute	Notes
1	**User-Name**	Used in access-request and account-request packets.
2	**User-Password**	—
3	**CHAP-Password**	*Not supported.*
4	**NAS-IP-Address**	Sent with every access-request. Specifies which switches a user may have access to. More than one of these attributes is allowed per user.
5	**NAS-Port**	Virtual port number sent with access-request and account-request packets. Slot/port information is supplied in attribute 26 (vendor-specific).
6 7 8 9 10 11 12 13 14 15 16	**Service-Type** **Framed-Protocol** **Framed-IP-Address** **Framed-IP-Netmask** **Framed-Routing** **Filter-Id** **Framed-MTU** **Framed-Compression** **Login-IP-Host** **Login-Service** **Login-TCP-Port**	*Not supported. These attributes are used for dial-up sessions; not applicable to the RADIUS client in the switch.*
17	**Unassigned**	—
18	**Reply-Message**	Multiple reply messages are supported, but the length of all the reply messages returned in one access-accept or access-reject packet cannot exceed 256 characters.

Num.	Standard Attribute	Notes
19 20 21 22 23	**Callback-Number** **Callback-Id** **Unassigned** **Frame-Route** **Framed-IPX-Network**	*Not supported. These attributes are used for dial-up sessions; not applicable to the RADIUS client in the switch.*
24	**State**	Sent in challenge/response packets.
25	**Class**	Used to pass information from the server to the client and passed unchanged to the accounting server as part of the accounting-request packet.
26	**Vendor-Specific**	See "Vendor-Specific Attributes for RADIUS" on page 20-11.
27	**Session-Timeout**	*Not supported.*
28	**Idle-Timeout**	*Not supported.*
29 30 31 32 33 34 35 36 37 38 39 60 61 62 63	**Termination-Action** **Called-Station-Id** **Calling-Station-Id** **NAS-Identifier** **Proxy-State** **Login-LAT-Service** **Login-LAT-Node** **Login-LAT-Group** **Framed-AppleTalk-Link** **Framed-AppleTalk-Network** **Framed-AppleTalk-Zone** **CHAP-Challenge** **NAS-Port-Type** **Port-Limit** **Login-LAT-Port**	*Not supported. These attributes are used for dial-up sessions; not applicable to the RADIUS client in the switch.*

Vendor-Specific Attributes for RADIUS

The Alcatel RADIUS client supports attribute 26, which includes a vendor ID and some additional subattributes called subtypes. The vendor ID and the subtypes collectively are called Vendor Specific Attributes (VSAs). Alcatel, through partnering arrangements, has included these VSAs in some vendors' RADIUS server configurations.

The attribute subtypes are defined in the server's dictionary file. If you are using single authority mode, the first VSA subtype, Alcatel-Auth-Vlan, must be defined on the server for each authenticated VLAN. Alcatel's vendor ID is 800 (SMI Network Management Private Enterprise Code).

The following are VSAs for RADIUS servers:

Num.	RADIUS VSA	Type	Description
1	**Alcatel-Auth-Group**	**integer**	The authenticated VLAN number. The only protocol associated with this attribute is Ethernet II. If other protocols are required, use the protocol attribute instead.
2	**Alcatel-Slot-Port**	**string**	Slot(s)/port(s) valid for the user.
3	**Alcatel-Time-of-Day**	**string**	The time of day valid for the user to authenticate.
4	**Alcatel-Client-IP-Addr**	**address**	The IP address used for Telnet only.
5	**Alcatel-Group-Desc**	**string**	Description of the authenticated VLAN.
6	**Alcatel-Port-Desc**	**string**	Description of the port.
8	**Alcatel-Auth-Group-Protocol**	**string**	The protocol associated with the VLAN. Must be configured for access to other protocols. Values include: **IP_E2**, **IP_SNAP**, **IPX_E2**, **IPX_NOV**, **IPX_LLC**, **IPX_SNAP**.
9	**Alcatel-Asa-Access**	**string**	Specifies that the user has access to the switch. The only valid value is **all**.
10	**Alcatel-End-User-Profile**	**string**	Specifies the name of an end-user profile associated with the user. Profiles are configured directly on the switch. See the *OmniSwitch 7700/7800/8800 Switch Management Guide* for more details.
39	**Alcatel-Acce-Priv-F-R1**	**hex.**	Configures functional read privileges for the user.
40	**Alcatel-Acce-Priv-F-R2**	**hex.**	Configures functional read privileges for the user.
41	**Alcatel-Acce-Priv-F-W1**	**hex.**	Configures functional write privileges for the user.
42	**Alcatel-Acce-Priv-F-W2**	**hex.**	Configures functional write privileges for the user.

The Alcatel-Auth-Group attribute is used for Ethernet II only. If a different protocol, or more than one protocol is required, use the Alcatel-Auth-Group-Protocol attribute instead. For example:

```
Alcatel-Auth-Group-Protocol 23: IP_E2 IP_SNAP
Alcatel-Auth-Group-Protocol 24: IPX_E2
```

In this example, authenticated users on VLAN 23 may use Ethernet II or SNAP encapsulation. Authenticated users on VLAN 24 may use IPX with Ethernet II.

Configuring Functional Privileges on the Server

Configuring the functional privileges attributes (**Alcatel-Acce-Priv-F-*x***) can be cumbersome because it requires using read and write bitmasks for command families on the switch.

1 To display the functional bitmasks of the desired command families, use the **show aaa priv hexa** command.

2 On the RADIUS server, configure the functional privilege attributes with the bitmask values.

Note. For information about configuring users on the switch, see the "Switch Security" chapter in the *OmniSwitch 7700/7800/8800 Switch Management Guide*.

RADIUS Accounting Server Attributes

The following table lists the standard attributes supported for RADIUS accounting servers. The attributes in the **radius.ini** file may be modified if necessary.

Num.	Standard Attribute	Description
1	**User-Name**	Used in access-request and account-request packets.
4	**NAS-IP-Address**	Sent with every access-request. Specifies which switches a user may have access to. More than one of these attributes is allowed per user.
5	**NAS-Port**	Virtual port number sent with access-request and account-request packets. Slot/port information is supplied in attribute 26 (vendor-specific).
25	**Class**	Used to pass information from the server to the client and passed unchanged to the accounting server as part of the accounting-request packet.
40	**Acct-Status-Type**	Four values should be included in the dictionary file: 1 (acct-start), 2 (acct-stop), 6 (failure), and 7 (acct-on). Start and stop correspond to login/logout. The accounting-on message is sent when the RADIUS client is started. This attribute also includes an accounting-off value, which is not supported.
42	**Acct-Input-Octets**	(Authenticated VLANs only) Tracked per port.
43	**Acct-Output-Octets**	(Authenticated VLANs only) Tracked per port.
44	**Acct-Session**	Unique accounting ID. (For authenticated VLAN users, Alcatel uses the client's MAC address.)
45	**Acct-Authentic**	Indicates how the client is authenticated; standard values (1–3) are not used. Vendor specific values should be used instead: AUTH-AVCLIENT (4) AUTH-TELNET (5) AUTH-HTTP (6) AUTH-NONE (0)
46	**Acct-Session**	The start and stop time for a user's session can be determined from the accounting log.
47	**Acct-Input-Packets**	(Authenticated VLANs only) Tracked per port.
48	**Acct-Output-Packets**	(Authenticated VLANs only) Tracked per port.
49	**Acct-Terminal-Cause**	Indicates how the session was terminated: NAS-ERROR USER-ERROR LOST CARRIER USER-REQUEST STATUS-FAIL

The following table lists the VSAs supported for RADIUS accounting servers. The attributes in the **radius.ini** file may be modified if necessary.

Num.	Accounting VSA	Type	Description
1	**Alcatel-Auth-Group**	**integer**	The authenticated VLAN number. The only protocol associated with this attribute is Ethernet II. If other protocols are required, use the protocol attribute instead.
2	**Alcatel-Slot-Port**	**string**	Slot(s)/port(s) valid for the user.
4	**Alcatel-Client-IP-Addr**	**dotted decimal**	The IP address used for Telnet only.
5	**Alcatel-Group-Desc**	**string**	Description of the authenticated VLAN.

Configuring the RADIUS Client

Use the **aaa radius-server** command to configure RADIUS parameters on the switch.

RADIUS server keywords	
key **host** **retransmit**	**timeout** **auth-port** **acct-port**

When creating a new server, at least one host name or IP address (specified by the **host** keyword) is required as well as the shared secret (specified by the **key** keyword).

In this example, the server name is **rad1**, the host address is 10.10.2.1, the backup address is 10.10.3.5, and the shared secret is **amadeus**. Note that the shared secret must be configured exactly the same as on the server.

```
-> aaa radius-server rad1 host 10.10.2.1 10.10.3.5 key amadeus
```

To modify a RADIUS server, enter the server name and the desired parameter to be modified.

```
-> aaa radius-server rad1 key mozart
```

If you are modifying the server and have just entered the **aaa radius-server** command to create or modify the server, you can use command prefix recognition. For example:

```
-> aaa radius-server rad1 retransmit 5
-> timeout 5
```

For information about server defaults, see “Server Defaults” on page 20-3.

To remove a RADIUS server, use the **no** form of the command:

```
-> no aaa radius-server rad1
```

Note that only one server may be deleted at a time.

LDAP Servers

Lightweight Directory Access Protocol (LDAP) is a standard directory server protocol. The LDAP client in the switch is based on several RFCs: 1798, 2247, 2251, 2252, 2253, 2254, 2255, and 2256. The protocol was developed as a way to use directory services over TCP/IP and to simplify the directory access protocol (DAP) defined as part of the Open Systems Interconnection (OSI) effort. Originally it was a front-end for X.500 DAP.

The protocol synchronizes and governs the communications between the LDAP client and the LDAP server. The protocol also dictates how its databases of information, which are normally stored in hierarchical form, are searched, from the root directory down to distinct entries.

In addition, LDAP has its own format that permits LDAP-enabled Web browsers to perform directory searches over TCP/IP.

Setting Up the LDAP Authentication Server

1 Install the directory server software on the server.

2 Copy the relevant schema LDIF files from the Alcatel software CD to the configuration directory on the server. (Each server type has a command line tool or a GUI tool for importing LDIF files.) Database LDIF files may also be copied and used as templates. The schema files and the database files are specific to the server type. The files available on the Alcatel software CD include the following:

aaa_schema.microsoft.ldif
aaa_schema.netscape.ldif
aaa_schema.novell.ldif
aaa_schema.openldap.schema
aaa_schema.sun.ldif

aaa_database.microsoft.ldif
aaa_database.netscape.ldif
aaa_database.novell.ldif
aaa_database.openldap.ldif
aaa_database.sun.ldif

3 After the server files have been imported, restart the server.

Note. Schema checking should be enabled on the server.

Information in the server files must match information configured on the switch through the **aaa ldap-server** command. For example, the port number configured on the server must be the same as the port number configured on the switch. See "Configuring the LDAP Authentication Client" on page 20-26 for information about using this command.

LDAP Server Details

LDAP servers must be configured with the properly defined LDAP schema and correct database suffix, including well-populated data. LDAP schema is extensible, permitting entry of user-defined schema as needed.

LDAP servers are also able to import and export directory databases using LDIF (LDAP Data Interchange Format).

LDIF File Structure

LDIF is used to transfer data to LDAP servers in order to build directories or modify LDAP databases. LDIF files specify multiple directory entries or changes to multiple entries, but not both. The file is in simple text format and can be created or modified in any text editor. In addition, LDIF files import and export binary data encoded according to the base 64 convention used with MIME (Multipurpose Internet Mail Extensions) to send various media file types, such as JPEG graphics, through electronic mail.

An LDIF file entry used to define an organizational unit would look like this:

```
dn: <distinguished name>
objectClass: top
objectClass: organizationalUnit
ou: <organizational unit name>
<list of optional attributes>
```

Below are definitions of some LDIF file entries:

entries	definition
dn: <distinguished name>	Defines the DN (required).
objectClass: top	Defines top object class (at least one is required). Object class defines the list of attributes required and allowed in directory server entries.
objectClass: organizationalUnit	Specifies that organizational unit should be part of the object class.
ou: <organizationalUnit name>	Defines the organizational unit's name.
<list of attritbutes>	Defines the list of optional entry attributes.

Common Entries

The most common LDIF entries describe people in companies and organizations. The structure for such an entry might look like the following:

```
dn: <distinguished name>
objectClass: top
objectClass: person
objectClass: organizational Person
cn: <common name>
sn: <surname>
<list of optional attributes>
```

This is how the entry would appear with actual data in it.

```
dn: uid=yname, ou=people, o=yourcompany
objectClass: top
objectClass: person
objectClass: organizational Person
cn: your name
sn: last name
givenname: first name
uid: yname
ou: people
description:
<list of optional attributes>
. . .
```

Directory Entries

Directory entries are used to store data in directory servers. LDAP–enabled directory entries contain information about an object (person, place, or thing) in the form of a Distinguished Name (DN) that should be created in compliance with the LDAP protocol naming conventions.

Distinguished names are constructed from Relative Distinguished Names (RDNs), related entries that share no more than one attribute value with a DN. RDNs are the components of DNs, and DNs are string representations of entry names in directory servers.

Distinguished names typically consist of descriptive information about the entries they name, and frequently include the full names of individuals in a network, their email addresses, TCP/IP addresses, with related attributes such as a department name, used to further distinguish the DN. Entries include one or more object classes, and often a number of attributes that are defined by values.

Object classes define all required and optional attributes (a set of object classes is referred to as a “schema”). As a minimum, every entry must include the DN and one defined object class, like the name of an organization. Attributes required by a particular object class must also be defined. Some commonly used attributes that comprise a DN include the following:

Country (c), State or Province (st), Locality (l),
Organization (o), Organization Unit (ou),
and Common Name (cn)

Although each attribute would necessarily have its own values, the attribute syntax determines what kind of values are allowed for a particular attribute, e.g., (c=US), where country is the attribute and US is the value. Extra consideration for attribute language codes will be necessary if entries are made in more than one language.

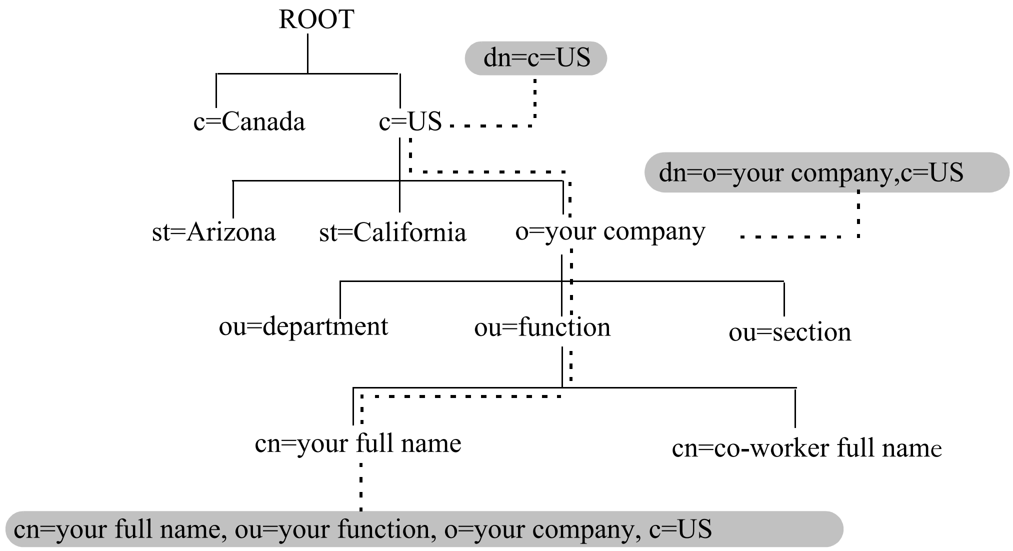
Entries are usually based on physical locations and established policies in a Directory Information Tree (DIT); the DN locates an entry in the hierarchy of the tree. Alias entries pointing to other entries can also be used to circumvent the hierarchy during searches for entries.

Once a directory is set up, DN attributes should thereafter be specified in the same order to keep the directory paths consistent. DN attributes are separated by commas as shown in this example:

```
cn=your name, ou=your function, o= your company, c=US
```

As there are other conventions used, please refer to the appropriate RFC specification for further details.

In addition to managing attributes in directory entries, LDAP makes the descriptive information stored in the entries accessible to other applications. The general structure of entries in a directory tree is shown in the following illustration. It also includes example entries at various branches in the tree.



Directory Information Tree

Directory Searches

DNs are always the starting point for searches unless indicated otherwise in the directory schema.

Searches involve the use of various criteria including scopes and filters which must be predefined, and utility routines, such as Sort. Searches should be limited in scope to specific durations and areas of the directory. Some other parameters used to control LDAP searches include the size of the search and whether to include attributes associated with name searches.

Base objects and scopes are specified in the searches, and indicate where to search in the directory. Filters are used to specify entries to select in a given scope. The filters are used to test the existence of object class attributes, and enable LDAP to emulate a “read” of entry listings during the searches. All search preferences are implemented by means of a filter in the search. Filtered searches are based on some component of the DN.

Retrieving Directory Search Results

Results of directory searches are individually delivered to the LDAP client. LDAP referrals to other servers are not returned to the LDAP client, only results or errors. If referrals are issued, the server is responsible for them, although the LDAP client will retrieve results of asynchronous operations.

Directory Modifications

Modifications to directory entries contain changes to DN entry attribute values, and are submitted to the server by an LDAP client application. The LDAP-enabled directory server uses the DNs to find the entries to either add or modify their attribute values.

Attributes are automatically created for requests to add values if the attributes are not already contained in the entries.

All attributes are automatically deleted when requests to delete the last value of an attribute are submitted. Attributes can also be deleted by specifying delete value operations without attaching any values.

Modified attribute values are replaced with other given values by submitting replace requests to the server, which then translates and performs the requests.

Directory Compare and Sort

LDAP will compare directory entries with given attribute values to find the information it needs. The Compare function in LDAP uses a DN as the identity of an entry, and searches the directory with the type and value of an attribute. Compare is similar to the Search function, but simpler.

LDAP will also sort entries by their types and attributes. For the Sort function, there are essentially two methods of sorting through directory entries. One is to sort by entries where the DN (Distinguished Name) is the sort key. The other is to sort by attributes with multiple values.

The LDAP URL

LDAP URLs are used to send search requests to directory servers over TCP/IP on the internet, using the protocol prefix: **ldap://.** (Searches over SSL would use the same prefix with an “s” at the end, i.e., **ldaps:**//.)

LDAP URLs are entered in the command line of any web browser, just as HTTP or FTP URLs are entered. When LDAP searches are initiated LDAP checks the validity of the LDAP URLs, parsing the various components contained within the URLs to process the searches. LDAP URLs can specify and implement complex or simple searches of a directory depending on what is submitted in the URLs. Searches performed directly with LDAP URLs are affected by the LDAP session parameters described above.

In the case of multiple directory servers, LDAP URLS are also used for referrals to other directory servers when a particular directory server does not contain any portion of requested IP address information. Search requests generated through LDAP URLs are not authenticated.

Searches are based on entries for attribute data pairs.

The syntax for TCP/IP LDAP URLs is as follows:

ldap://<hostname>:<port>/<base_dn>?attributes>?<scope>?<filter>

An example might be:

ldap://ldap.company name.xxx/o=company name%inc./,c=US>
(base search including all attributes/object classes in scope).

LDAP URLs use the percent symbol to represent commas in the DN. The following table shows the basic components of LDAP URLs.

components	description
<ldap>	Specifies TCP/IP connection for LDAP protocol. (The <ldaps> prefix specifies SSL connection for LDAP protocol.)
<hostname>	Host name of directory server or computer, or its IP address (in dotted decimal format).
<port>	TCP/IP port number for directory server. If using TCP/IP and default port number (389), port need not be specified in the URL. SSL port number for directory server (default is 636).
<base_dn>	DN of directory entry where search is initiated.
<attributes>	Attributes to be returned for entry search results. All attributes are returned if search attributes are not specified.
<scope>	Different results are retrieved depending on the scopes associated with entry searches. "base" search: retrieves information about distinguished name as specified in URL. This is a <base_dn> search. Base searches are assumed when the scope is not designated. "one" (one-level) search: retrieves information about entries one level under distinguished name (<base_dn> as specified in the URL, excluding the base entry. "sub" (subtree) search: retrieves information about entries from all levels under the distinguished name (<base_dn>) as specified in the URL, including the base entry.
<filter>	Search filters are applied to entries within specified search scopes. Default filter objectClass=* is used when filters are not designated. (Automatic search filtering not yet available.)

Password Policies and Directory Servers

Password policies applied to user accounts vary slightly from one directory server to another. Normally, only the password changing policies can be set by users through the directory server graphical user interface (GUI). Other policies accessible only to Network Administrators through the directory server GUI may include one or more of the following operational parameters.

- Log-in Restrictions
- Change Password
- Check Password Syntax
- Password Minimum Length
- Send Expiration Warnings
- Password History
- Account Lockout

- Reset Password Failure Count
- LDAP Error Messages (e.g., Invalid Username/Password, Server Data Error, etc.)

For instructions on installing LDAP-enabled directory servers, refer to the vendor-specific instructions.

Directory Server Schema for LDAP Authentication

Object classes and attributes will need to be modified accordingly to include LDAP authentication in the network (object classes and attributes are used specifically here to map user account information contained in the directory servers).

- All LDAP-enabled directory servers require entry of an auxiliary objectClass:passwordObject for user password policy information.
- Another auxiliary objectClass: password policy is used by the directory server to apply the password policy for the entire server. There is only one entry of this object for the database server.

Note. Server schema extensions should be configured before the **aaa ldap-server** command is configured.

Vendor-Specific Attributes for LDAP Servers

The following are Vendor Specific Attributes (VSAs) for Authenticated Switch Access and/or Layer 2 Authentication:

attribute	description
bop-asa-func-priv-read-1	Read privileges for the user.
bop-asa-func-priv-read-2	Read privileges for the user.
bop-asa-func-priv-write-1	Write privileges for the user.
bop-asa-func-priv-write-2	Write privileges for the user.
bop-asa-allowed-access	Whether the user has access to configure the switch.
bop-asa-snmp-level-security	Whether the user may have SNMP access, and the type of SNMP protocol used.
bop-shakey	A key computed from the user password with the alp2key tool.
bop-md5key	A key computed from the user password with the alp2key tool.
allowedtime	The periods of time the user is allowed to log into the switch.
switchgroups	The VLAN ID and protocol (**IP_E2**, **IP_SNAP**, **IPX_E2**, **IPX_NOV**, **IPX_LLC**, **IPX_SNAP).**

Configuring Functional Privileges on the Server

Configuring the functional privileges attributes (**bop-asa-func-priv-read-1**, **bop-asa-func-priv-read-2**, **bop-asa-func-priv-write-1**, **bop-asa-func-priv-write-2**) requires using read and write bitmasks for command families on the switch.

1 To display the functional bitmasks of the desired command families, use the **show aaa priv hexa** command.

2 On the LDAP server, configure the functional privilege attributes with the bitmask values.

For more information about configuring users on the switch, see the Switch Security chapter of the *OmniSwitch 7700/7800/8800 Switch Management Guide*.

Configuring Authentication Key Attributes

The alp2key tool is provided on the Alcatel software CD for computing SNMP authentication keys.The alp2key application is supplied in two versions, one for Unix (Solaris 2.5.1 or higher) and one for Windows (NT 4.0 and higher).

To configure the bop-shakey or bop-md5key attributes on the server:

1 Use the alp2key application to calculate the authentication key from the password of the user. The switch automatically computes the authentication key, but for security reasons the key is never displayed in the CLI.

2 Cut and paste the key to the relevant attribute on the server.

An example using the alp2key tool to compute the SHA and MD5 keys for **mypassword**:

```
ors40595{}128: alp2key mypassword
bop-shakey: 0xb1112e3472ae836ec2b4d3f453023b9853d9d07c
bop-md5key: 0xeb3ad6ba929441a0ff64083d021c07f1
ors40595{}129:
```

Note. The bop-shakey and bop-md5key values must be recomputed and copied to the server any time a user’s password is changed.

LDAP Accounting Attributes

Logging and accounting features include Account Start, Stop and Fail Times, and Dynamic Log. Typically, the Login, Logout and Dynamic logs can be accessed from the directory server software. Additional third-party software is required to retrieve and reset the log information to the directory servers for billing purposes.

The following sections describe accounting server attributes.

AccountStartTime

User account start times are tracked in the AccountStartTime attribute of the user's directory entry that keeps the time stamp and accounting information of user log-ins. The following fields (separated by carriage returns "|") are contained in the Login log. Some fields are only used for Layer 2 Authentication.

Fields Included For Any Type of Authentication

- User account ID or username client entered to log-in: variable length digits.
- Time Stamp (YYYYMMDDHHMMSS (YYYY:year, MM:month, DD:day, HH:hour, MM:minute, SS:second)
- Switch serial number: Alcatel.BOP.<*switch name*>.<*MAC address*>
- Client IP address: variable length digits.

Fields Included for Layer 2 Authentication Only

- Client MAC address: xx:xx:xx:xx:xx:xx:xx (alphanumeric).
- Switch VLAN number client joins in multiple authority mode (0=single authority; 2=multiple authority); variable-length digits.
- Switch slot number to which client connects: nn
- Switch port number to which client connects: nn
- Switch virtual interface to which client connects: nn

AccountStopTime

User account stop times are tracked in the AccountStopTime attribute that keeps the time stamp and accounting information of successful user log-outs. The same fields as above (separated by carriage returns "|") are contained in the Logout log. A different carriage return such as the # sign may be used in some situations. Additionally, these fields are included but apply only to the Logout log:

Fields For Any Type of Authentication

- Log-out reason code, for example LOGOFF(18) or DISCONNECTED BY ADMIN(19)
- User account ID or username client entered to log-in: variable length digits.

Fields For Layer 2 Authentication Only

- Number of bytes received on the port during the client's session from log-in to log-out: variable length digits.
- Number of bytes sent on the port during the client's session from log-in to log-out: variable length digits.

- Number of frames received on the port during the client's session from log-in to log-out: variable length digits.
- Number of frames sent on the port during the clients session from log-in to log-out: variable length digits.

AccountFailTime

The AccountFailTime attribute log records the time stamp and accounting information of unsuccessful user log-ins. The same fields in the Login Log—which are also part of the Logout log (separated by carriage returns "|")—are contained in the Login Fail log. A different carriage return such as the # sign may be used in some situations. Additionally, these fields are included but apply only to the Login Fail log.

- User account ID or username client entered to log-in: variable length digits.
- Log-in fail error code: nn. For error code descriptions refer to the vendor-specific listing for the specific directory server in use.
- Log-out reason code, for example PASSWORD EXPIRED(7) or AUTHENTICATION FAILURE(21)

Dynamic Logging

Dynamic logging may be performed by an LDAP-enabled directory server if an LDAP server is configured **first** in the list of authentication servers configured through the the **aaa accounting vlan** or **aaa accounting session** command. Any other servers configured are used for accounting (storing history records) only. For example:

```
-> aaa accounting session ldap2 rad1 rad2
```

In this example, server **ldap2** will be used for dynamic logging, and servers **rad1** and **rad2** will be used for accounting.

If you specify a RADIUS server first, all of the servers specified will be used for recording history records (not logging). For example:

```
-> aaa accounting session rad1 ldap2
```

In this example, both the **rad1** and **ldap2** servers will be used for history only. Dynamic logging will not take place on the LDAP server.

Dynamic entries are stored in the LDAP-enabled directory server database from the time the user successfully logs in until the user logs out. The entries are removed when the user logs out.

- Entries are associated with the switch the user is logged into.
- Each dynamic entry contains information about the user's connection. The related attribute in the server is bop-loggedusers.

A specific object class called **alcatelBopSwitchLogging** contains three attributes as follows:

Attribute	Description
bop-basemac	MAC range, which uniquely identifies the switch
bop-switchname	Host name of the switch.
bop-loggedusers	Current activity records for every user logged onto the switch identified by bop-basemac.

Each switch that is connected to the LDAP-enabled directory server will have a DN starting with bop-basemac-*xxxxx*, ou=bop-logging. If the organizational unit ou=bop.logging exists somewhere in the tree under searchbase, logging records are written on the server. See the server manufacturer's documentation for more information about setting up the server.

The bop-loggedusers attribute is a formatted string with the following syntax:

loggingMode : accessType ipAddress port macAddress vlanList userName

The fields are defined here:

Field	Possible Values
loggingMode	**ASA** *x*—for an authenticated user session, where *x* is the number of the session **AVLAN**—for Authenticated VLAN session in single authority mode **AVLAN** *y*—for Authenticated VLAN session in multiple authority mode, where *y* is relevant VLAN
accessType	Any one of the following: **CONSOLE**, **MODEM**, **TELNET**, **HTTP**, **FTP**, **XCAP**
ipAddress	The string **IP** followed by the IP address of the user.
port	(For Authenticated VLAN users only.) The string **PORT** followed by the slot/port number.
macAddress	(For Authenticated VLAN users only.) The string **MAC** followed by the MAC address of the user.
vlanList	(For Authenticated VLAN users only.) The string **VLAN** followed by the list of VLANs the user is authorized (for single-mode authority).
userName	The login name of the user.

For example:

```
“ASA      0   :  CONSOLE IP 65.97.233.108   Jones”
```

Configuring the LDAP Authentication Client

Use the **aaa ldap-server** command to configure LDAP authentication parameters on the switch. The server name, host name or IP address, distinguished name, password, and the search base name are required for setting up the server. Optionally, a backup host name or IP address may be configured, as well as the number of retransmit tries, the timeout for authentication requests, and whether or not a secure Socket Layer (SSL) is enabled between the switch and the server.

Note. The server should be configured with the appropriate schema before the **aaa ldap-server** command is configured.

The keywords for the **aaa ldap-server** command are listed here:

Required for creating:	optional:
host **dn** **password** **base**	**type** **retransmit** **timeout** **port** **ssl**

Creating an LDAP Authentication Server

An example of creating an LDAP server:

```
-> aaa ldap-server ldap2 host 10.10.3.4 dn cn=manager password tpub base c=us
```

In this example, the switch will be able to communicate with an LDAP server (called **ldap2**) that has an IP address of 10.10.3.4, a domain name of cn=manager, a password of tpub, and a searchbase of c=us. These parameters must match the same parameters configured on the server itself.

Note. The distinguished name must be different from the searchbase name.

Modifying an LDAP Authentication Server

To modify an LDAP authentication server, use the **aaa ldap-server** command with the server name; or, if you have just entered the **aaa ldap-server** command to create or modify the server, you can use command prefix recognition. For example:

```
-> aaa ldap-server ldap2 password my_pass
-> timeout 4
```

In this example, an existing LDAP server is modified with a different password, and then the timeout is modified on a separate line. These two command lines are equivalent to:

```
-> aaa ldap-server ldap2 password my_pass timeout 4
```

Setting Up SSL for an LDAP Authentication Server

A Secure Socket Layer (SSL) may be set up on the server for additional security. When SSL is enabled, the server's identity will be authenticated. The authentication requires a certificate from a Certification Authority (CA). If the CA providing the certificate is well-known, the certificate is automatically extracted from the **Fbase.img** file on the switch (**certs.pem**). If the CA is not well-known, the CA's certificate must be transfered to the switch via FTP to the /flash/certified or /flash/working directory and should be named **optcerts.pem**. The switch merges either or both of these files into a file called **ldapcerts.pem**.

To set up SSL on the server, specify **ssl** with the **aaa ldap-server** command:

```
-> aaa ldap-server ldap2 ssl
```

The switch automatically sets the port number to 636 when SSL is enabled. The 636 port number is typically used on LDAP servers for SSL. The port number on the switch must match the port number configured on the server. If the port number on the server is different from the default, use the **aaa ldap-server** command with the **port** keyword to configure the port number. For example, if the server port number is 635, enter the following:

```
-> aaa ldap-server ldap2 port 635
```

The switch will now be able to communicate with the server on port 635.

To remove SSL from the server, use **no** with the **ssl** keyword. For example:

```
-> aaa ldap-server ldap2 no ssl
```

SSL is now disabled for the server.

Removing an LDAP Authentication Server

To delete an LDAP server from the switch configuration, use the **no** form of the command with the relevant server name.

```
-> no aaa ldap-server topanga5
```

The topanga5 server is removed from the configuration.

Verifying the Authentication Server Configuration

To display information about authentication servers, use the following command:

show aaa server	Displays information about a particular AAA server or AAA servers.
show aaa priv hexa	Displays hexadecimal values for command domains/families.

An example of the output for this command is given in "Quick Steps For Configuring Authentication Servers" on page 20-4. For more information about the output of this command, see the *OmniSwitch CLI Reference Guide*.

21 Configuring Authenticated VLANs

Authenticated VLANs control user access to network resources based on VLAN assignment and a user log-in process; the process is sometimes called user authentication or Layer 2 Authentication. (Another type of security is device authentication, which is set up through the use of port-binding VLAN policies or static port assignment. See Chapter 8, "Defining VLAN Rules.") In this chapter, the terms *authenticated VLANs* (AVLANs) and *Layer 2 Authentication* are synonymous.

Layer 2 Authentication is different from another feature in the switch called Authenticated Switch Access, which is used to grant individual users access to manage the switch. For more information about Authenticated Switch Access, see the "Switch Security" chapter in the *Switch Management Guide*.

In This Chapter

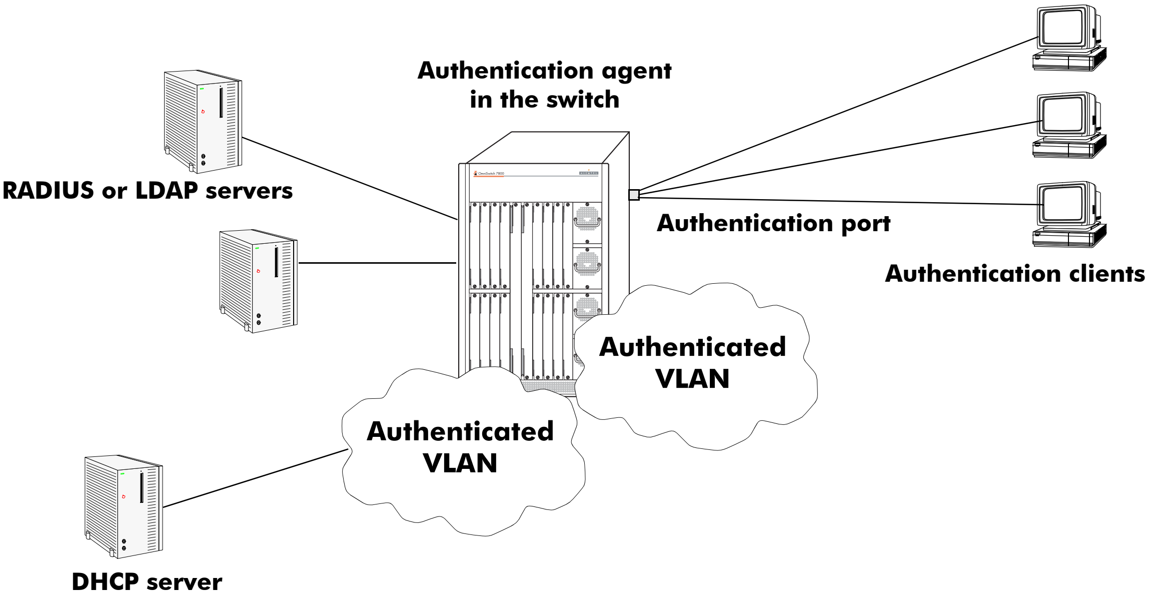
This chapter describes authenticated VLANs and how to configure them through the Command Line Interface (CLI). CLI commands are used in the configuration examples; for more details about the syntax of commands, see the *OmniSwitch CLI Reference Guide*.

The authentication components described in this chapter include:

- **Authentication clients**—see "Setting Up Authentication Clients" on page 21-7.
- **Authenticated VLANs**—see "Configuring Authenticated VLANs" on page 21-26.
- **Authentication ports**—see "Configuring Authenticated Ports" on page 21-28.
- **DHCP server**—see "Setting Up the DHCP Server" on page 21-29.
- **Authentication server authority mode**—see "Configuring the Server Authority Mode" on page 21-32.
- **Accounting servers**—see "Specifying Accounting Servers" on page 21-35.

Authenticated Network Overview

An authenticated network involves several components as shown in this illustration.



Authentication Network Components

This chapter describes all of these components in detail, except the external authentication servers, which are described in Chapter 20, "Managing Authentication Servers." A brief overview of the components is given here:

Authentication servers—A RADIUS or LDAP server must be configured in the network. The server contains a database of user information that the switch checks whenever a user tries to authenticate through the switch. (*Note that the local user database on the switch may not be used for Layer 2 authentication.*) Backup servers may be configured for the authentication server.

- **RADIUS or LDAP server**. Follow the manufacturer's instructions for your particular server. The external server may also be used for Authenticated Switch Access. Server details, such as RADIUS attributes and LDAP schema information, are given in Chapter 20, "Managing Authentication Servers."
- **RADIUS or LDAP client in the switch**. The switch must be set up to communicate with the RADIUS or LDAP server. This chapter briefly describes the switch configuration. See Chapter 20, "Managing Authentication Servers," for detailed information about setting up switch parameters for authentication servers.

Authentication clients—Authentication clients login through the switch to get access to authenticated VLANs. There are three types of clients:

- **AV-Client**. This is a proprietary authentication client. The AV-Client does not require an IP address prior to authentication. The client software must be installed on the user's end station. This chapter describes how to install and configure the client. See "Installing the AV-Client" on page 21-12.
- **Telnet client**. Any standard Telnet client may be used. An IP address is required prior to authentication. An overview of the Telnet client is provided in "Setting Up Authentication Clients" on page 21-7.

- **Web browser client**. Any standard Web browser may be used (Netscape or Internet Explorer). An IP address is required prior to authentication. See "Web Browser Authentication Client" on page 21-8 for more information about Web browser clients.

Authenticated VLANs—At least one authenticated VLAN must be configured. See "Configuring Authenticated VLANs" on page 21-26.

Authentication port—At least one mobile port must be configured on the switch as an authentication port. This is the physical port through which authentication clients are attached to the switch. See "Configuring Authenticated Ports" on page 21-28

DHCP Server—A DHCP server can provide IP addresses to clients prior to authentication. After authentication, any client can obtain an IP address in an authenticated VLAN to which the client is allowed access. A relay to the server must be set up on the switch. See "Setting Up the DHCP Server" on page 21-29.

Authentication agent in the switch—Authentication is enabled when the server(s) and the server authority mode is specified on the switch. See "Configuring the Server Authority Mode" on page 21-32.

These components are described in more detail in the next sections.

AVLAN Configuration Overview

Configuring authenticated VLANs requires several major steps. The steps are outlined here and described throughout this chapter. See “Sample AVLAN Configuration” on page 21-5 for a quick overview of implementing the commands used in these procedures.

1 Set up authentication clients. See “Setting Up Authentication Clients” on page 21-7.

2 Configure at least one authenticated VLAN. A router port must be set up in at least one authenticated VLAN for the DHCP relay. See “Configuring Authenticated VLANs” on page 21-26.

3 Configure at least one authenticated mobile port. Required for connecting the clients to the switch. See “Configuring Authenticated Ports” on page 21-28.

4 Set up the DHCP server. Required if you are using Telnet or Web browser clients. Required for any clients that need to get IP addresses after authentication. See “Setting Up the DHCP Server” on page 21-29.

5 Configure the authentication server authority mode. See “Configuring the Server Authority Mode” on page 21-32.

6 Specify accounting servers for authentication sessions. Optional; accounting may also be done through the switch logging feature in the switch. See “Specifying Accounting Servers” on page 21-35.

The following is a summary of commands used in these procedures.

Commands	Used for ...
vlan authentication	Enabling authentication on VLAN(s)
ip interface	Setting up a router port on the authenticated VLAN.
vlan port mobile **vlan port authenticate**	Creating authenticated port(s)
aaa avlan dns	Configuring a DNS name; required for Web browser clients
ip helper address **aaa avlan default dhcp** **ip helper avlan only**	Configuring the DHCP server; required for for Telnet and Web browser clients.
aaa vlan no	Removing a user from an authenticated VLAN
aaa ldap-server **aaa radius-server**	Setting up switch communication with authentication servers
aaa authentication vlan single-mode **aaa authentication vlan multiple-mode**	Enabling authentication and setting the authority mode for servers
aaa accounting vlan	Specifying accounting for AVLAN sessions.

Sample AVLAN Configuration

1 Enable at least one authenticated VLAN:

```
-> vlan 2 authentication enable
```

Note that this command does not create a VLAN; the VLAN must already be created. For information about creating VLANs, see Chapter 5, "Configuring VLANs."

The VLAN must also have a router port if Telnet or Web browser clients will be authenticating into this VLAN. The following command configures a router port on VLAN 2:

```
-> vlan 2 router ip 10.10.2.20
```

The switch will automatically create an authentication IP address based on this router port address (in this example, the address would be 10.10.2.253). The authentication address is configurable. See "Configuring Authentication IP Addresses" on page 21-27.

2 Create and enable at least one mobile authenticated port. The port must be in VLAN 1, the default VLAN on the switch.

```
-> vlan port mobile 3/1
-> vlan port 3/1 authenticate enable
```

3 Set up a DNS path if users will be authenticating through a Web browser:

```
-> aaa avlan dns auth.company
```

4 Set up a path to a DHCP server if users will be getting IP addresses from DHCP. The IP helper address is the IP address of the DHCP server; the AVLAN default DHCP address is the address of any router port configured on the VLAN.

```
-> ip helper address 10.10.2.5
-> aaa avlan default dhcp 10.10.2.20
```

If the relay will be used for authentication only, enter the **ip helper avlan only** command:

```
-> ip helper avlan only
```

Note. To check the DNS and DHCP authentication configuration, enter the **show aaa avlan config** command. For example:

```
-> show aaa avlan config
default DHCP relay address= 192.9.33.222
authentication DNS name    = authent.company.com
default traffic            = disabled
port bounding              = disabled
```

For more information about this command, see the *OmniSwitch CLI Reference Guide*.

---

5 Configure the switch to communicate with the authentication servers. Use the **aaa radius-server** or **aaa ldap-server** command. For example:

```
-> aaa radius-server rad1 host 10.10.1.2 key wwwtoe timeout 3
-> aaa ldap server ldap2 host 199.1.1.1 dn manager password foo base c=us
```

Each server may have another host configured as a backup host. See Chapter 20, "Managing Authentication Servers," for more information about setting up external servers for authentication.

6 Enable authentication by specifying the authentication mode (single mode or multiple mode) and the server. Use the RADIUS or LDAP server name(s) configured in step 5. For example:

```
-> aaa authentication vlan single-mode rad1 rad2
```

7 Set up an accounting server (for RADIUS or LDAP) for authentication sessions.

```
-> aaa accounting vlan rad3 local
```

Note. Verify the authentication server configuration by entering the **show aaa authentication vlan** command or verify the accounting server configuration by entering the **show aaa accounting vlan** command. For example:

```
-> show aaa authentication vlan
All authenticated vlans
1rst authentication server  = rad1,
2nd authentication server   = ldap2

-> show aaa accounting vlan
All authenticated vlans
1rst authentication server  = rad3,
2nd authentication server   = local
```

For more information about these commands, see the *OmniSwitch CLI Reference Guide*.

Setting Up Authentication Clients

The following sections describe the Telnet authentication client, Web browser authentication client, and Alcatel's proprietary AV-Client. For information about removing a particular client from an authenticated network, see "Configuring Authenticated VLANs" on page 21-26.

An overview of authentication clients is given in the following table:

Type of Client	Secure	Single Sign-on	IP Address Required	IP Release/ Renew	Platforms Supported
AV-Client	no	yes	no	automatic	Windows only (except ME)
Telnet	no	no	yes	manual	Windows Linux Mac OS 9.x (no Telnet by default) Mac OS X.1
Web Browser (HTTP)	yes (SSL)	no	yes	automatic	Windows (IE version 4.72 and later; Netscape version 4.7 and later) Linux (Netscape version 4.75 and later) Mac OS 9.x (IE versions 5.5 and later, including 5.0 and 5.14) Mac OS X.1 (IE versions between 5.0 and 5.5, except 5.0, 5.5, and 5.14)

Note. Secure Shell cannot be used for Authenticated VLAN sessions. To establish a secure AVLAN connection, use a Web Browser (HTTP) client.

Telnet Authentication Client

Telnet clients authenticate through a Telnet session.

- **Make sure a Telnet client is available on the client station**. No specialized authentication client software is required on Telnet client workstations.
- **Provide an IP address for the client**. Telnet clients require an address prior to authentication. The address may be statically assigned if the authentication network is set up in single authority mode with one authenticated VLAN. The address may be assigned dynamically if a DHCP server is located in the network. DHCP relay is required in networks with multiple authenticated VLANs.
- **Configure a DHCP server**. Telnet clients may get IP addresses via a DHCP server prior to authenticating or after authentication in order to move into a different VLAN. When multiple authenticated VLANs are configured, after the client authenticates the client must issue a DHCP release/renew request in order to be moved into the correct VLAN. Typically Telnet clients cannot automatically do a release/renew and must be manually configured. For information about configuring the DHCP server, see "Setting Up the DHCP Server" on page 21-29.

Web Browser Authentication Client

Web browser clients authenticate through the switch via any standard Web browser software (Netscape Navigator or Internet Explorer).

- **Make sure a standard browser is available on the client station**. No specialized client software is required.
- **Provide an IP address for the client**. Web browser clients require an address prior to authentication. The address may be statically assigned if the authentication network is set up in single authority mode with one authenticated VLAN. The address may be assigned dynamically if a DHCP server is located in the network. DHCP is required in networks with multiple authenticated VLANs.
- **Configure a DHCP server**. Web browser clients may get IP addresses via a DHCP server prior to authenticating or after authentication in order to move into a different VLAN. When multiple authenticated VLANs are configured, after the client authenticates the client must issue a DHCP release/renew request in order to be moved into the correct VLAN. Web browser clients automatically issue DHCP release/renew requests after authentication. For more information, see "Setting Up the DHCP Server" on page 21-29.
- **Configure a DNS name on the switch**. A DNS name must be configured so that users may enter a URL rather than an IP address in the browser command line. For more information, see "Setting Up a DNS Path" on page 21-29.

Configuring the Web Browser Client Language File

If you want the Web browser client to display the username and password prompts in another language, modify the label.txt file with the desired prompts.

The label.txt file is available in the **/flash/switch** directory when you install the **Fsecu.img** or **Esecu.img** file as described in the next section.

The file may be edited with any text editor, and the format of the username and password prompts is as follows:

Username="*username_string*"
Password="*password_string*"

Use the **aaa avlan http language** command to enable this file. For example:

```
-> aaa avlan http language
```

The label.txt file will be used for Web browser authentication clients.

Note. If you want to return to the default language (English) for the Web browser prompts, delete the contents of the file.

Installation Files for Web Browser Clients

Make sure the **/flash/switch/avlan** directory is available on the switch. The directory must be manually installed using the **install** command to load **Fsecu.img** (OmniSwitch 7700/7800) or **Esecu.img** (OmniSwitch 8800). The file is available in the working directory on the switch. When the file is installed, the /flash/switch/avlan directory will be available on the switch.

Important. When you install the **Fsecu.img** or **Esecu.img** file after initial installation, any files in the flash/switch/avlan directory will be overwritten.

The /flash/switch/avlan directory contains authentication HTML pages for the client that may be modified (to include a company logo, for example). The names of these files are: **topA.html**, **topB.html**, **bottomA.html**, **bottomB.html**, and **myLogo.gif**.

The directory also contains files that *must* be installed on Mac OS Web browser clients as described in the next sections.

Installing Files for Mac OS 9.x Clients

1 In the browser URL command line, enter the authentication DNS name (configured through the **aaa avlan dns** command). The authentication page displays.

2 Click on the link to download the installation software. The **javlanInstall.sit** file is copied to the Mac desktop.

3 Double-click the **javlanInstall.sit** file on the desktop.

4 Double-click on the application javlanInstall AppleScript inside the newly created directory. The workstation is now setup for authentication.

Installing Files for Mac OSX.1 Clients

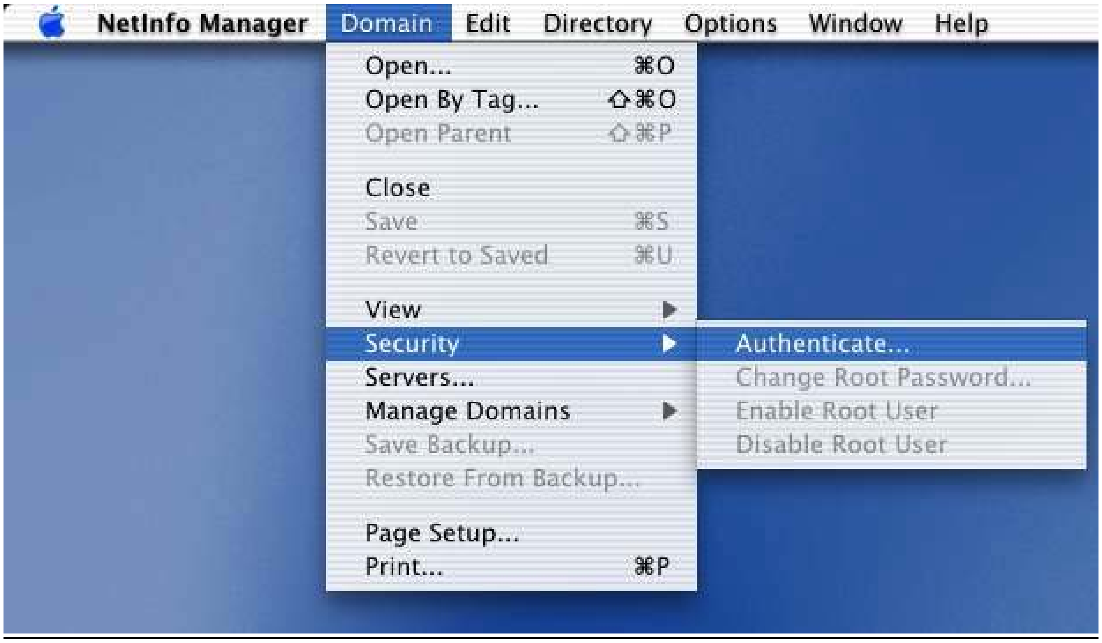
The installation must be done at the root. Root access is not automatic in OSX.1. A password must be set to activate it.

Disconnect the Mac’s network connection before setting root access. Otherwise, the NetInfo Manager application in the Mac OS will send multiple DNS requests, and the process to set root access will take longer.

To set root access:

1 Open the NetInfo from the HardDisk/Application/Utilities folder.

2 Select Domain > Security > Authenticate. Enter the administrator's password if required.



3 Select Domain > Security > Enable Root. Enter the password.

4 Select System Preferences/Login and select the login prompt to display when opening a new session.

5 Quit the current session and relogon as the root user.

6 Make sure Ethernet-DCHP is selected in the Network Utility.

7 Reconnect the Ethernet cable.

8 If you are using a self-signed SSL certificate, or the certificate provided by Alcatel (**wv-cert.pem**), see "DNS Name and Web Browser Clients" on page 21-11.

To set up the Mac OSX.1 for authentication:

1 In the browser URL command line, enter the DNS name configured on the switch (see the next section for setting up the DNS name for Mac OSX clients). The authentication page displays.

2 Click on the link to download the installation software. The **avlanInstall.tar** file is copied to the Mac desktop.

3 Double-click on the **avlanInstall.tar** file.

4 Make sure that Java is enabled in the browser application.

5 Make sure the SSL certificate is installed correctly (see "SSL for Web Browser Clients" on page 21-11) and that the DNS name configured on the switch matches the DNS name in the certificate (see "DNS Name and Web Browser Clients" on page 21-11).

SSL for Web Browser Clients

A Secure Socket Layer (SSL) is used to authenticate Web browser clients. A certificate from a Certification Authority (CA) or a self-signed (private) certificate must be installed on the switch. A self-signed certificate is provided by Alcatel (**wv-cert.pem**). If you are using a well-known certificate or some other self-signed certificate, you should replace the **wv-cert.pem** file with the relevant file.

Web browser clients will automatically recognize well-known SSL certificates, but if a self-signed certificate (such as the **wv-cert.pem** file) is used, the client will not automatically recognize the certificate.

Windows, Linux, and Mac OS 9 Clients

If you are using the **wv-cert.pem** file or another self-signed certificate, the client will not recognize the certificate, and a warning message will display on the client; however, the client will be allowed to authenticate.

Mac OSX.1 Clients

On Mac OSX.1, if you are using the **wv-cert.pem** file or another self-signed certificate, the certificate file must be FTP'd to the workstation and installed with the **keytool** command as follows:

1 FTP the **wv-cert.pem** file (or the relevant certificate file) from the /flash/switch directory on the switch to the workstation.

2 On the Mac workstation, open a Terminal application at the root (see the previous section for information about enabling root access). Enter the following command:

keytool -import -keystore <*path to JDK installation*>/lib/security/cacerts -alias ALCATEL_AVLAN - file <*path to certificate file*>

For example:

```
keytool -import -keystore /System/Library/Frameworks/JavaVM.framework/Versions/
1.3.1/Home/lib/security/cacerts -alias ALCATEL_AVLAN - file/Users/endalat/
Destop/wv-cert.pem
```

Note. The **keytool** command requires a password. By default, the password is **changeit**.

DNS Name and Web Browser Clients

For Mac OSX.1 clients, the DNS name in the certificate must match the DNS name configured on the switch through the **aaa avlan dns** command. If the DNS names do not match, the Java applet in the client cannot be loaded and the client cannot authenticate. (For other clients, if the DNS names do not match, a warning will display when the client attempts to authenticate; however, the client is still allowed to authenticate.)

The **wv-cert.pem** certificate contains a default DNS name (**webview**). To configure the DNS name on the switch, enter the **aaa avlan dns** command with the DNS name matching the one in the certificate. For example:

```
-> aaa dns avlan webview
```

On the browser workstation, the authentication user must enter the DNS name in the browser command line to display the authentication page.

For more information about configuring a DNS name, see "Setting Up a DNS Path" on page 21-29.

Installing the AV-Client

The AV-Client is a proprietary Windows-based application that is installed on client end stations. The installation instructions are provided in this chapter.

The AV-Client does not require an IP address in order to authenticate; the client relies on the DLC protocol (rather than IP) to communicate with the authentication agent in the switch. After authentication, the client may issue a DHCP release/renew request to get an IP address; a utility in the client software may be used to configure this automatic request. For information about configuring the utility, see "Configuring the AV-Client Utility" on page 21-18.

The AV-Client software requires three main installation steps as listed here. These steps are slightly different depending on the version of Windows you are using.

- **Load the Microsoft DLC protocol stack**. See "Loading the Microsoft DLC Protocol Stack" on page 21-12.
- **Load the AV-Client software**. See "Loading the AV-Client Software" on page 21-14.
- **Set the AV-Client as primary network login (Windows 95 and 98)**. See "Setting the AV-Client as Primary Network Login" on page 21-18.
- **Configure the AV-Client for DHCP (optional)**. See "Configuring the AV-Client Utility" on page 21-18.

Loading the Microsoft DLC Protocol Stack

Windows 2000 and Windows NT

You must have the DLC protocol installed on your Windows PC workstation before you install the AV-Client. The installation of the DLC protocol stack may require files from the Windows distribution software. Make sure to have your Windows media available during this procedure. Follow these steps to load the protocol on a Windows workstation.

1. From your Windows desktop, select Start > Settings > Control Panel.
2. Double-click the Network icon. When the Network window opens, select the Protocols tab.
3. Click the **Add** button and the Select Network Protocol window appears.
4. Select the DLC protocol from the list of Network Protocols. Click **OK.**
5. Follow the screen prompts requesting Windows files.

Windows 98

1. From your Windows desktop, select Start > Settings > Control Panel.
2. Double-click the Network icon. When the Network window opens, select the Configuration tab.
3. Click the **Add** button and the Select Network Component Type window appears.
4. Select Protocol and click the **Add** button.
5. When the Select Network Protocol window appears, select Microsoft from the list of manufacturers and Microsoft 32-bit DLC from the list of Network Protocols. Click **OK**.
6. Follow the prompts requesting Windows files.

Windows 95

Install the 32-bit DLC protocol program and the update patch from the Microsoft FTP site (ftp.microsoft.com). From the FTP site, download the MSDLC32.EXE and DLC32UPD.EXE files (or the latest DLC protocol update). These files are self-extracting zip files. Follow these steps:

1 Double-click the MSDLC32.EXE file in the folder to which you want to download the file.

Note. Do not run MSDLC32.EXE file in the Windows or Windows/System folders. If you downloaded the file to either of these locations, copy it to a temporary folder on your hard disk or copy it to an installation diskette before double-clicking on it.

2 From your Windows desktop, select Start > Settings > Control Panel.

3 Double-click the Network icon in the Control Panel.

4 In the Network dialog box, click on the **Add** button.

5 In the Select Network Component Type dialog box, double-click on the Protocol network component.

6 In the Select Network Protocol dialog box, click on the **Have Disk** button.

7 Specify the drive and path where the MSDLC32.EXE files (you should have already extracted them) are located. For example, if you created an installation diskette, you would enter

```
<drive letter>:\
```

If you created a temporary folder on your hard disk, then you would enter

```
C:\<folder name>
```

where folder name is the directory or path into which you copied the MSDLC32.EXE files. Click **OK**.

8 Click “Microsoft 32-bit DLC”, then click **OK** again.

9 When prompted, insert the Windows 95 disks so that other network components can be reinstalled.

10 When prompted, shut down your computer and restart Windows 95. This restart is required for the DLC protocol stack to load on the system.

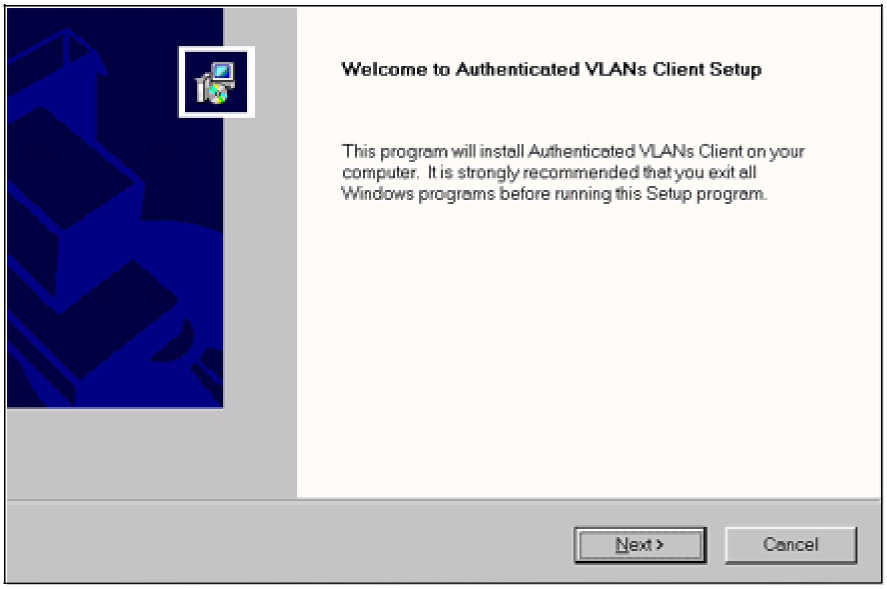
11 Next, the DLC protocol stack update must be loaded. Double click the DLC32UPD.EXE file. The program will install itself. After installing the update, it is recommended that the system be rebooted.

Loading the AV-Client Software

Windows 2000 and Windows NT

1 Download the AV-Client from the Alcatel website onto the Windows desktop.

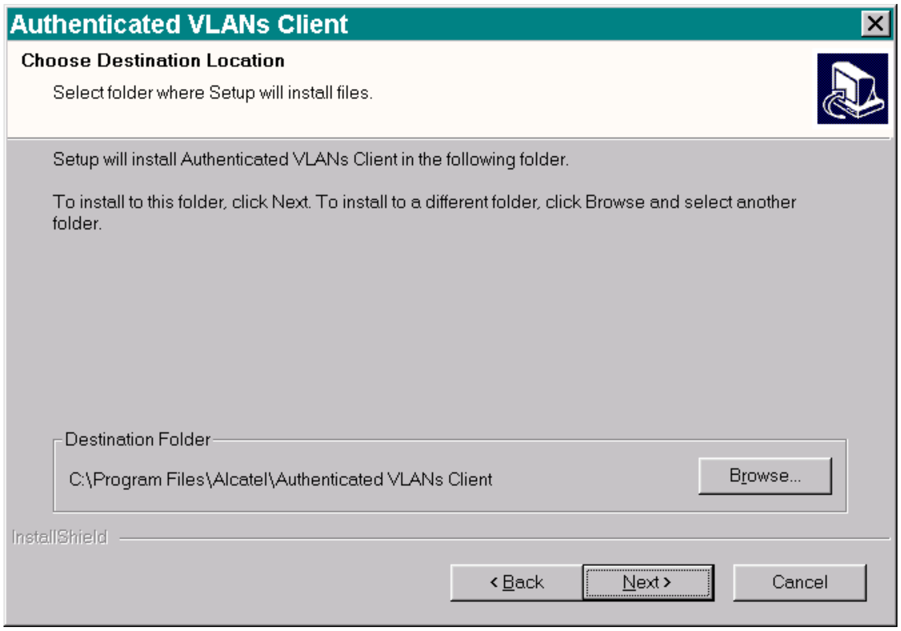
2 Double-click the AV-Client icon. The installation routine begins and the following window displays:



Welcome to Authenticated VLANs Client Setup

This program will install Authenticated VLANs Client on your computer. It is strongly recommended that you exit all Windows programs before running this Setup program.

3 We recommend that you follow the instructions on the screen regarding closing all Windows programs before proceeding with the installation. Click on the **Next** button. The following window displays.



Authenticated VLANs Client

Choose Destination Location

Select folder where Setup will install files.

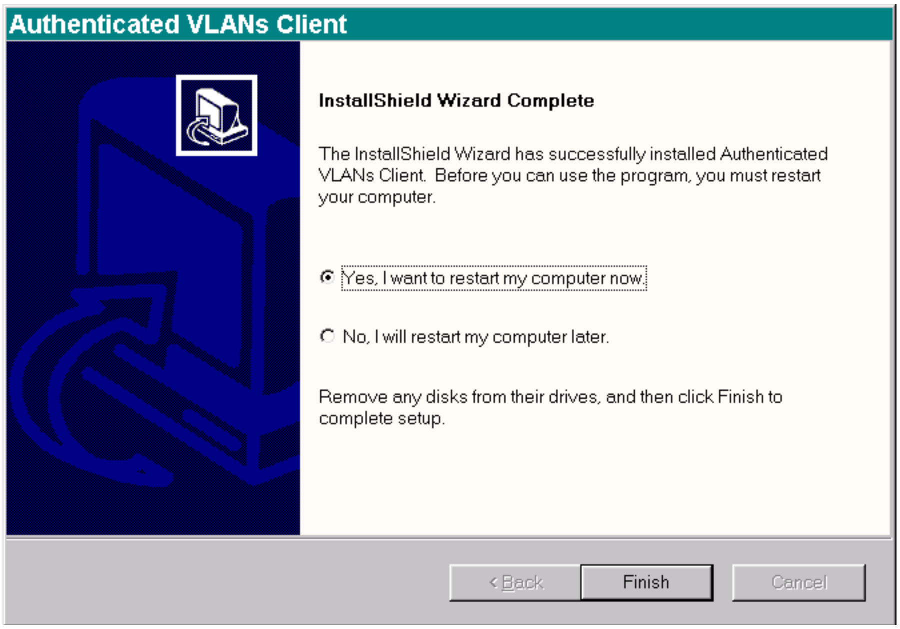
Setup will install Authenticated VLANs Client in the following folder.

To install to this folder, click Next. To install to a different folder, click Browse and select another folder.

Destination Folder

C:\Program Files\Alcatel\Authenticated VLANs Client

4 From this window you may install the client at the default destination folder shown on the screen or you may click the **Browse** button to select a different directory. Click on the **Next** button. The software loads, and the following window displays.



Authenticated VLANs Client

InstallShield Wizard Complete

The InstallShield Wizard has successfully installed Authenticated VLANs Client. Before you can use the program, you must restart your computer.

● Yes, I want to restart my computer now.

○ No, I will restart my computer later.

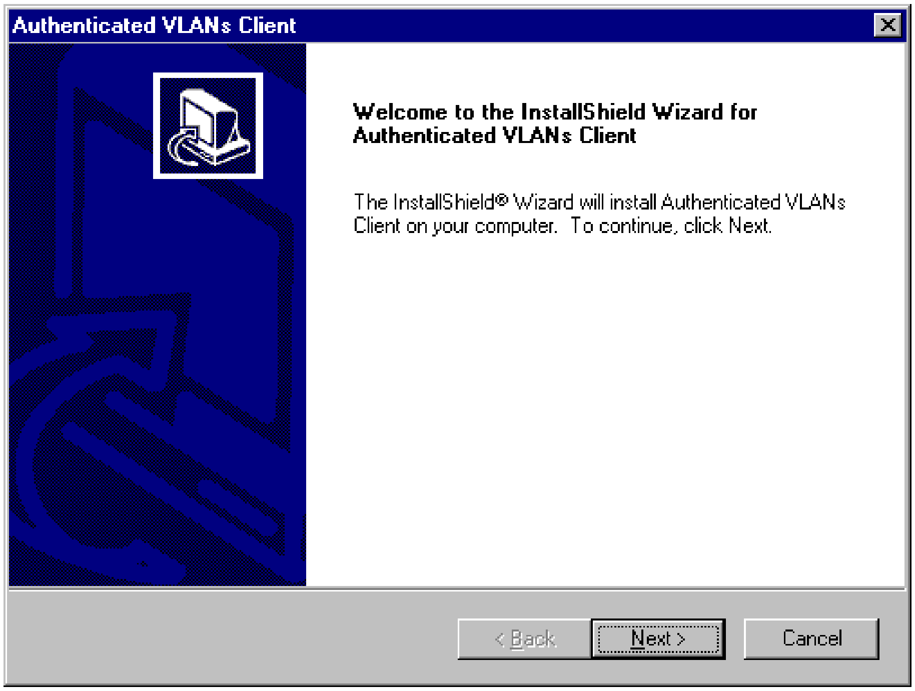
Remove any disks from their drives, and then click Finish to complete setup.

5 This window gives you the option of restarting your PC workstation now, or later. You cannot use the AV-Client until you restart your computer. If you decide to restart now, be sure to remove any disks from their drives. Click the **Finish** button to end the installation procedure.

Windows 95 and Windows 98

1 Download the AV-Client from the Alcatel website onto the Windows desktop.

2 Double-click the AV-Client icon. The installation routine begins and the following window displays:



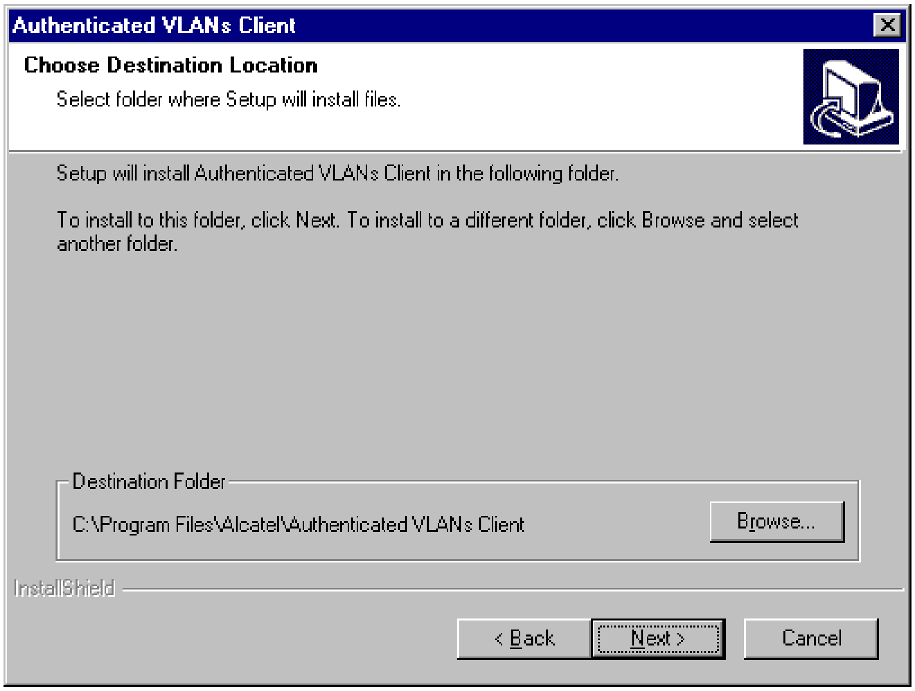
Authenticated VLANs Client

Welcome to the InstallShield Wizard for Authenticated VLANs Client

The InstallShield® Wizard will install Authenticated VLANs Client on your computer. To continue, click Next.

3 We recommend that you follow the instructions on the screen regarding closing all Windows programs before proceeding with the installation. Click on the **Next** button. The following window displays:

.



Authenticated VLANs Client

Choose Destination Location

Select folder where Setup will install files.

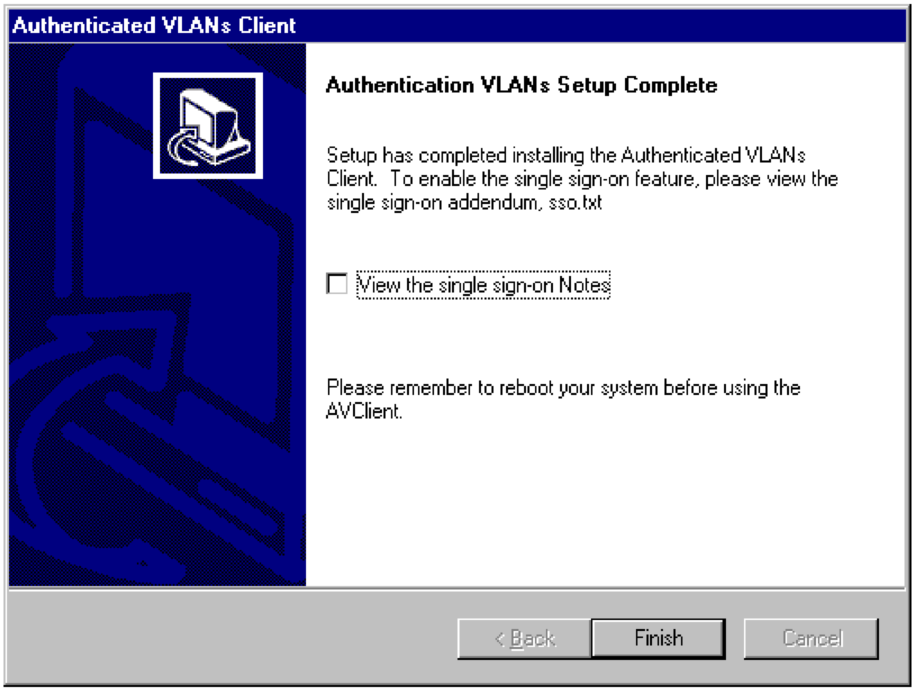
Setup will install Authenticated VLANs Client in the following folder.

To install to this folder, click Next. To install to a different folder, click Browse and select another folder.

Destination Folder

C:\Program Files\Alcatel\Authenticated VLANs Client

4 From this window you may install the client at the default destination folder shown on the screen or you may click the **Browse** button to select a different directory. Click on the **Next** button. The software loads, and the following window displays.



Authenticated VLANs Client

Authentication VLANs Setup Complete

Setup has completed installing the Authenticated VLANs Client. To enable the single sign-on feature, please view the single sign-on addendum, sso.txt

□ View the single sign-on Notes

Please remember to reboot your system before using the AVClient.

5 This window recommends that you read a text file included with the client before you exit the install shield. Click on the box next to “View the single sign-on Notes” to select this option. Click on the **Finish** button to end the installation process. Remember that you must restart your computer before you can run the AV-Client.

Setting the AV-Client as Primary Network Login

Windows 95 and Windows 98

If your operating system is Windows 95 or Windows 98, you must configure the AV-Client as the primary network login. This is done via the Windows Control Panel. From your Windows desktop, select Start > Settings > Control Panel. Double-click on the Network icon on the Control Panel window. From the Configuration Tab, proceed as follows:

1 Click the **Add** button.

2 Select the “Client” from the list and click the **Add** button. The “Select Network Client Window” displays.

3 You can click the **Have Disk** button, enter the correct path for your disk drive in the space provided and click **OK**. You can also browse to the directory where the AV-Client is installed and click **OK**. Select “Alcatel AVLAN Login Provider”.

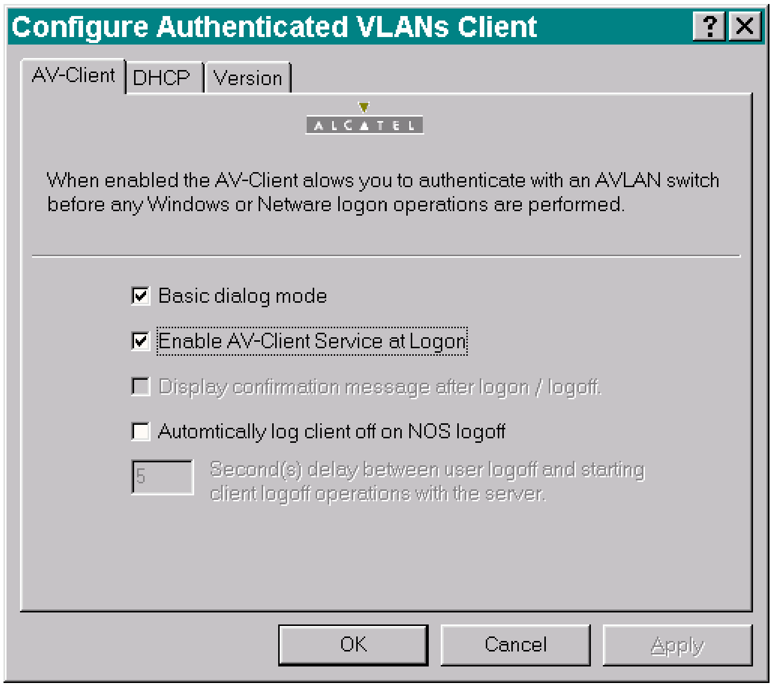
4 Select Alcatel AVLAN Login Provider as the Primary Network Login on the Configuration tab.

5 Complete the setup as prompted by Windows.

Note. Make sure to have your Windows 95 or 98 media available during this procedure.

Configuring the AV-Client Utility

The AV-Client includes a utility for configuring client options. To run the utility, install the AV-Client and reboot the PC workstation. From your Windows desktop, select Start > Settings > Control Panel. Double-click on the Authenticated VLANs Client icon in the Control Panel window. You can also access the utility by pointing your mouse to the AV-Client icon on the Windows system tray and executing a right click to select **Settings**. The following screen displays:



When enabled the AV-Client allows you to authenticate with an AVLAN switch before any Windows or Netware logon operations are performed.

☑ Basic dialog mode

☑ Enable AV-Client Service at Logon

☐ Display confirmation message after logon / logoff.

☐ Automtically log client off on NOS logoff

5 Second(s) delay between user logoff and starting client logoff operations with the server.

Selecting a Dialog Mode

The AV-Client has two dialog modes, basic and extended. In basic dialog mode, the client prompts the user for a username and a password only. In extended mode, which is required for multiple authority authentication, the client login screen also prompts the user for a VLAN number and optional challenge code. These additional authentication parameters are defined when the authentication server is configured in multiple authority mode.

You can set the dialog mode from the AV-Client's Control Panel Window. The basic dialog mode is enabled by default. To enable extended mode, de-select basic mode by clicking "Basic dialog mode." The **Apply** button will activate. Click the **Apply** button. The next time the AV-Client is started extended mode will be enabled.

Enabling/disabling the AV-Client at Startup

1 To enable/disable the AV-Client at startup, from your Windows desktop, select Start, Settings, Control Panel to access the AV-Client configuration utility. Select the AV-Client tab.

2 Click on the box next to "Enable AV-Client Service at Logon." The check mark in the box will disappear and the **Apply** button will activate.

3 To apply the change, click the **Apply** button. When you click the **OK** button, the screen will close, the change will take effect and the AV-Client will be disabled at logon. If you decide not to implement the change, click the **Cancel** button and the screen will close.

Note. If you disable the AV-Client at startup, you can activate VLAN authentication by pointing your mouse to the AV-Client icon on the Windows stem tray and right-clicking to select Logon.

Automatic Client or NOS Logoff

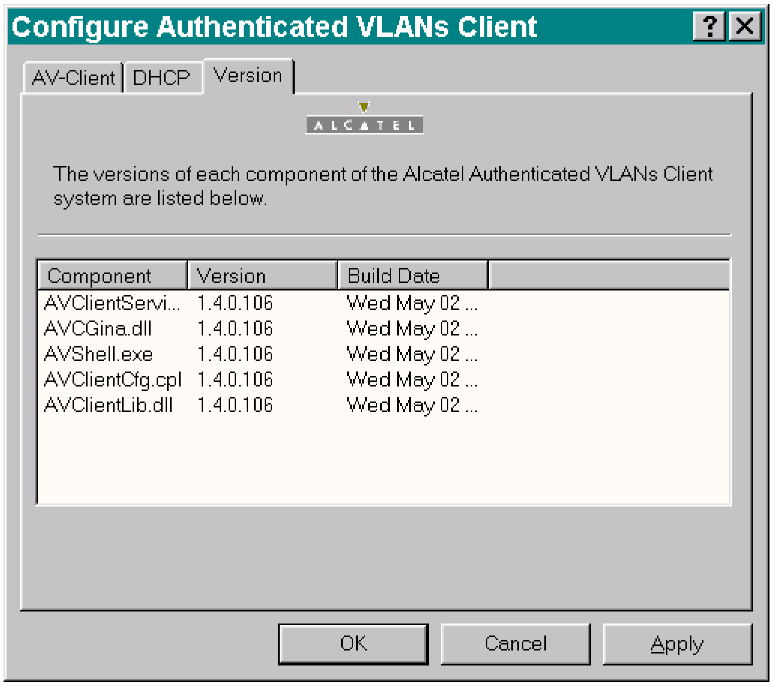
The default configuration of the client is to logoff the authentication client when the user logs off the desktop. You can configure the client so the workstation is automatically logged off when the user logs off.

To set this option, access the AV-Client configuration utility and click the box next to the "Automatically log client off or NOS logoff" option. When the option activates, you then have the option of setting a time delay between the moment the user logs off the workstation and the moment the client logs out of server operations.

Note. If the user reboots the PC workstation, the client's session with the network server is automatically terminated.

Viewing AV-Client Components

The configuration utility includes a screen that lists each component, version and build date for the AV-Client. To view this screen, click on the Version tab and a screen similar to the following will display.



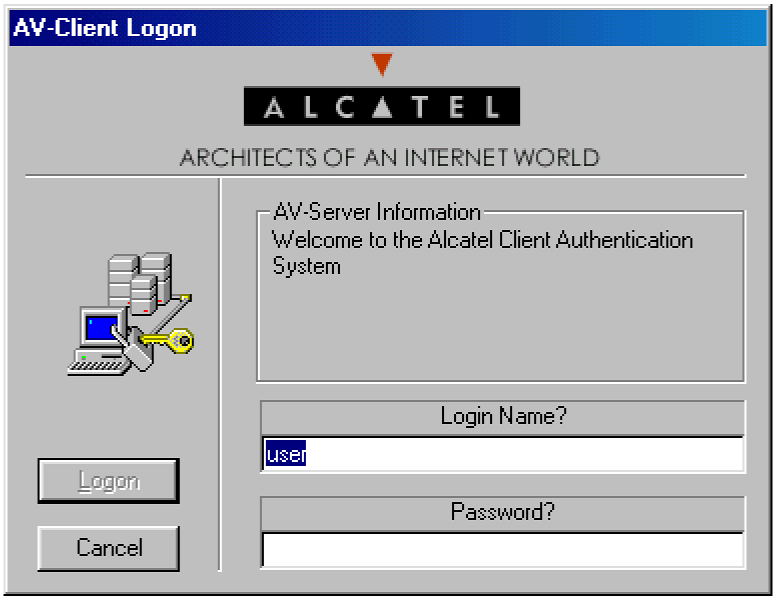
The versions of each component of the Alcatel Authenticated VLANs Client system are listed below.

Component	Version	Build Date
AVClientServi...	1.4.0.106	Wed May 02 ...
AVCGina.dll	1.4.0.106	Wed May 02 ...
AVShell.exe	1.4.0.106	Wed May 02 ...
AVClientCfg.cpl	1.4.0.106	Wed May 02 ...
AVClientLib.dll	1.4.0.106	Wed May 02 ...

Logging Into the Network Through an AV-Client

Once the AV-Client software has been loaded on a user's PC workstation, an AV-Client icon will be created on the Windows desktop in the task bar. Follow these steps to log into the authentication network:

1 Right click the AV-Client icon and select Logon. The following login screen displays:



AV-Server Information
Welcome to the Alcatel Client Authentication System

Login Name? user

Password?

2 Enter the user name for this device in the "Login Name?" field. This user name is configured on the authentication server.

3 Enter the password for this user in the "Password?" field. If the client is set up for basic dialog mode and the user enters the correct password, the user is authenticated. If the client is set up for extended mode, the user will be prompted to enter the VLAN ID and challenge. After all required user information is entered, the following message displays:

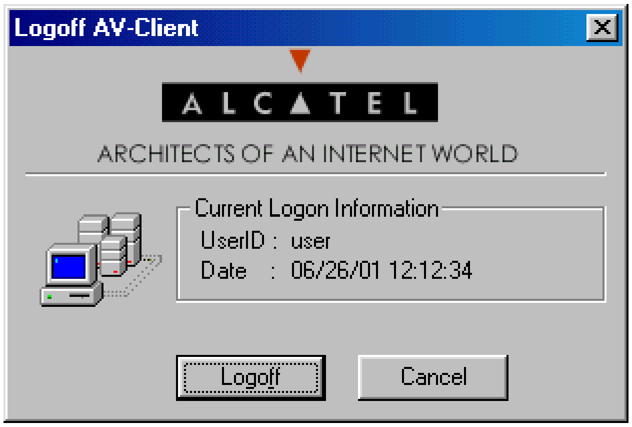
```
User xxxx authenticated by <Authentication Type> authentication
```

The user is now logged into the network and has access to all network resources in the VLAN with which this user shares membership.

Note. If authentication is successful but an error was made while configuring VLANs, the user station may not move into the VLAN the user requested.

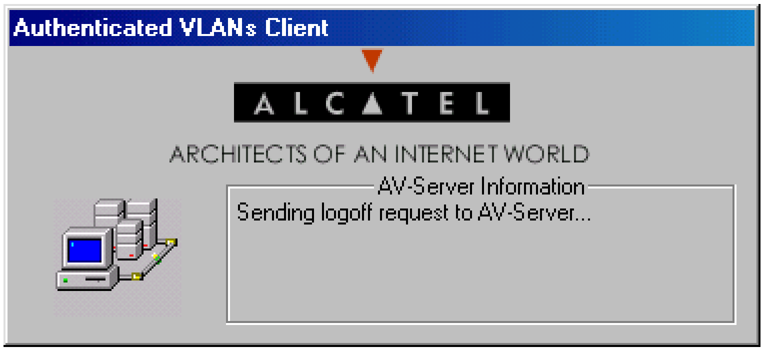
Logging Off the AV-Client

1 To log off the AV-Client, point your mouse to the AV-Client icon in your Windows system tray and execute a right-click to select Logoff. The following screen displays.



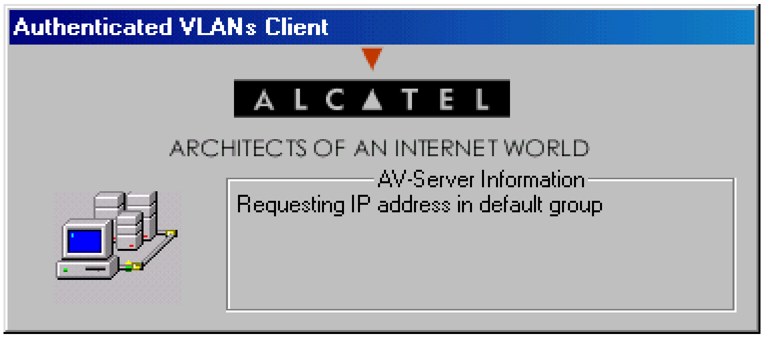
Current Logon Information
UserID : user
Date : 06/26/01 12:12:34

2 To continue the procedure, click the **Logoff** button. The following screen indicates that the AV-Client is sending a logoff request to the authentication server.



AV-Server Information
Sending logoff request to AV-Server...

The next message on the screen indicates that the AV-Client is requesting an IP address in the default VLAN. The client is removed from the authenticated VLAN and placed in the default VLAN.



AV-Server Information
Requesting IP address in default group

When the AV-Client is logged into the network, the AV-Client icon on the Windows desktop has a blue background. When the logoff procedure is completed, the screen disappears and the background is gone from the AV-Client icon.

Configuring the AV-Client for DHCP

For an AV-Client, DHCP configuration is not required. AV-Clients do not require an IP address to authenticate, but they may want an IP address for IP communication in an authenticated VLAN.

Note. If the AV-Client will be used with DHCP, the DHCP server must be configured as described in "Setting Up the DHCP Server" on page 21-29.

At startup, an AV-Client user PC workstation will issue a Windows DHCP request if the AV-Client's DHCP release/renew feature is enabled. This feature is disabled by default. The AV-Client is capable of obtaining an address from the default client VLAN or whatever VLAN it authenticates into if a DHCP server is located in the VLAN.

The DHCP tab of the configuration utility gives you several options for managing DHCP when it is enabled. You also have the option of disabling DHCP operations.

Delay for IP Address Request

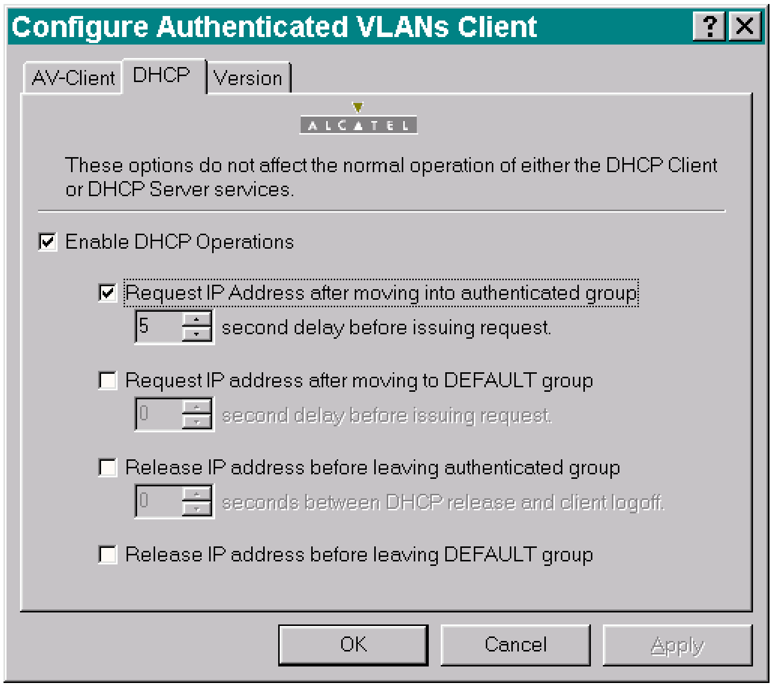
- You can specify a delay between the moment the client workstation moves into an authentication VLAN and the moment a DHCP request is issued for an IP address.
- You can specify a delay between the moment the client workstation moves into the default VLAN and the moment a DHCP request is issued for an IP address.

Releasing the IP Address

- You can specify a delay between the moment the client workstation logs off the network and the DHCP releases the IP address assigned to the client.
- You can configure the utility so that DHCP releases the IP address before the client workstation leaves the default VLAN.

Note. A delay between DHCP release and client logoff is recommended because the DHCP server's MAC address may be timed out in the AV-Client's ARP table. If that is the case, the client must send an ARP packet to discover the DHCP server's MAC address before it can send the release packet. If the logoff packet is sent to the switch before the release packet gets sent, then the IP address will never be released. Increasing the value of the delay parameter can prevent this from happening.

1 To configure the DHCP parameters, access the AV-Client configuration utility and select the DHCP tab. The following screen displays:



These options do not affect the normal operation of either the DHCP Client or DHCP Server services.

☑ Enable DHCP Operations

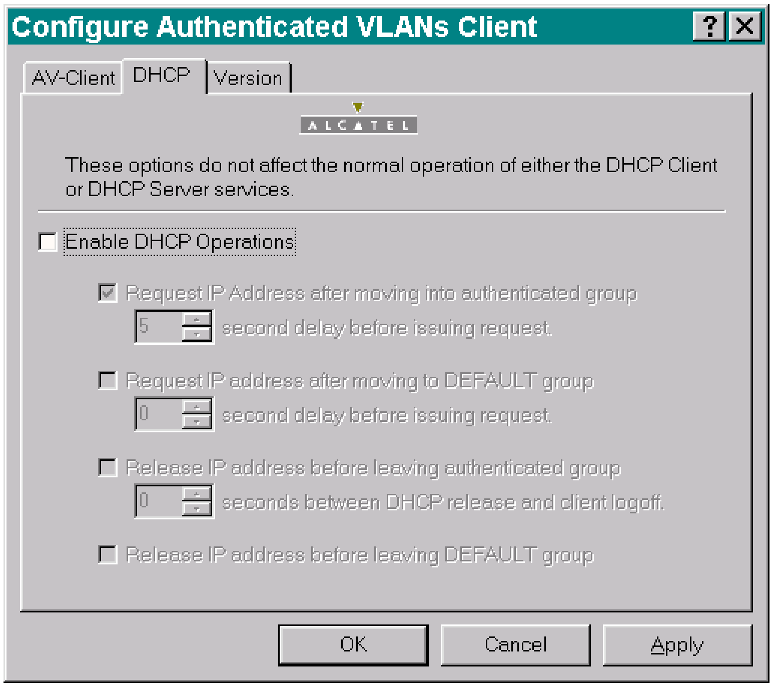
☑ Request IP Address after moving into authenticated group
5 second delay before issuing request.

□ Request IP address after moving to DEFAULT group
0 second delay before issuing request.

□ Release IP address before leaving authenticated group
0 seconds between DHCP release and client logoff.

□ Release IP address before leaving DEFAULT group

2 Click the box next to "Enable DHCP Operations". Several options will activate in the utility window as shown in the following screen. When you click on a box next to an option, the option is activated in the configuration window.



These options do not affect the normal operation of either the DHCP Client or DHCP Server services.

□ Enable DHCP Operations

☑ Request IP Address after moving into authenticated group
5 second delay before issuing request.

□ Request IP address after moving to DEFAULT group
0 second delay before issuing request.

□ Release IP address before leaving authenticated group
0 seconds between DHCP release and client logoff.

□ Release IP address before leaving DEFAULT group

3 When you click one of the features, an indicator is activated directly below the feature. Specify the number of seconds for the delay for the selected feature.

4 To apply the change, click the **Apply** button. When you click the **OK** button, the screen will close and the change will take effect. If you decide not to implement the change, click the **Cancel** button and the screen will close without implementing a change.

Configuring Authenticated VLANs

At least one authenticated VLAN must be configured on the switch. For more information about VLANs in general, see Chapter 5, “Configuring VLANs.”

To configure an authenticated VLAN, use the **vlan authentication** command to enable authentication on an existing VLAN. For example:

```
-> vlan 2 authentication enable
```

Note that the specified VLAN (in this case, VLAN 2) must already exist on the switch. A router port must also be configured for the VLAN (with the **ip interface** command) so that a DHCP relay may be set up. For example:

```
-> vlan 2 router ip 10.10.2.20
```

See “Setting Up the DHCP Server” on page 21-29 for more information about setting up a DHCP server.

Removing a User From an Authenticated Network

To remove a user from authenticated VLANs, enter the **aaa vlan no** command with the user’s MAC address. If the user’s MAC address is unknown, enter the **show avlan user** command first. Specify the VLAN ID or slot number to get information about a particular VLAN or slot only. For example:

```
-> show avlan user 23
name              Mac Address           Slot    Port   Vlan

-----------------------------------------------------------------
user1              00:20:da:05:f6:23    02       02      23
```

In this example, user1 is authenticated into VLAN 23 and is using MAC address 00:20:da:05:f6:23. To remove user1 from authenticated VLAN 23, enter the **aaa vlan no** command with the MAC address. For example:

```
-> aaa avlan no 00:20:da:05:f6:23
```

When this command is entered, user1 will be removed from VLAN 23. If the switch is set up so that authenticated users may traffic in the default VLAN, the user will be placed into the default VLAN of the authentication port. (See “Setting Up the Default VLAN for Authentication Clients” on page 21-27 for information about setting up the switch so that authentication clients may traffic in the default VLAN prior to authentication.)

For more information about the output display for the **aaa avlan no** and **show avlan user** commands, see the *OmniSwitch CLI Reference Guide*.

Note. The MAC addresses of users may also be found in the log files generated by accounting servers.

Configuring Authentication IP Addresses

Authentication clients connect to an IP address on the switch for authentication. (Web browser clients may enter a DNS name rather than the IP address; see "Setting Up a DNS Path" on page 21-29). When the router port is set up for an authenticated VLAN (through the **ip interface** command), the switch automatically sets up an authentication address for that authenticated VLAN based on the router port address. The authentication address uses the same mask as the router port address and includes *.253* at the end of the address.

For example, if the router port address for authenticated VLAN 3 is 10.10.2.20, the authentication address will be 10.10.2.253. This address is modifiable through the **avlan auth-ip** command; the address, however, must use the same mask as the router port address. For example:

```
-> avlan auth-ip 3 10.10.2.80
```

This changes the authentication address for VLAN 3 to 10.10.2.80. The authentication IP address is also used for the DNS address (see "Setting Up a DNS Path" on page 21-29).

To display authentication addresses, use the **show aaa avlan auth-ip** command.

Setting Up the Default VLAN for Authentication Clients

By default, authentication users cannot traffic in the default VLAN prior to authentication; however, the switch may be configured to enable the default VLAN so that users may traffic in the default VLAN prior to authentication.

The default VLAN is the default VLAN for the authentication port, the physical port through which authentication clients are connected to the switch. The authentication port is specified through the **vlan port authenticate** command. See "Configuring Authenticated Ports" on page 21-28.

Use the **avlan default-traffic** command to enable the default VLAN for authentication traffic.

```
-> avlan default-traffic enable
```

When this command is enabled, any authentication client initially belongs to the default VLAN of the authentication port through which the client is connected. After authentication, if a client is removed from an authenticated VLAN through the **aaa avlan no** command, the client is moved to the default VLAN.

To disable any default VLAN for authentication traffic, use the **disable** keyword with the command:

```
-> avlan default-traffic disable
WARNING: Traffic on default vlan is DISABLED.
         Existing users on default vlan are not flushed.
```

Users now do not belong to and cannot traffic in the default VLAN prior to authentication. Note that any existing users in the default VLAN are not flushed.

Port Binding and Authenticated VLANs

By default, authenticated VLANs do not support port binding rules. These rules are used for assigning devices to authenticated VLANs when device traffic coming in on an authenticated port matches criteria specified in the rule.

You can globally enable the switch so that port binding rules may be enabled on any authenticated VLAN on the switch.

The port binding rule types that are allowed on authenticated VLANs are as follows:

- MAC-Port-IP address
- MAC-Port
- Port-IP address
- MAC-Port-Protocol

The MAC-IP address and Port-Protocol binding rules are not supported on authenticated VLANs. For more information about port binding rules and how to configure them, see Chapter 8, “Defining VLAN Rules.”

To enable port binding rules on authenticated VLANs, use the **avlan port-bound** command with the **enable** keyword.

```
-> avlan port-bound enable
```

This command allows some port binding rules (MAC-Port-IP address, MAC-Port, Port-IP address, and MAC-Port-Protocol) to be used on any authenticated VLAN.

To disable port binding rules on authenticated VLANs, use the **disable** keyword with the command:

```
-> avlan port-bound disable
```

This command disables port binding rules on all authenticated VLANs.

Configuring Authenticated Ports

At least one mobile port must be configured as the physical port through which authentication clients connect to the switch.

To create a mobile port, use the **vlan port mobile** command.

```
-> vlan port mobile 3/1
```

To enable authentication on the mobile port, use the **vlan port authenticate** command:

```
-> vlan port 3/1 authenticate enable
```

For more information about configuring VLAN ports, see Chapter 7, “Assigning Ports to VLANs.”

By default, authentication clients cannot traffic in the default VLAN for the authentication port unless the **avlan default-traffic** command is enabled. See “Setting Up the Default VLAN for Authentication Clients” on page 21-27.

Setting Up a DNS Path

A Domain Name Server (DNS) name may be configured so that Web browser clients may enter a URL on the browser command line instead of an authentication IP address. A Domain Name Server must be set up in the network for resolving the name to the authentication IP address.

There may be multiple authentication IP addresses on the switch (if multiple authenticated VLANs are set up); however, there is only one authentication DNS path or host name. When the client enters the DNS path, the switch determines the IP authentication address based on the client's IP address, and the browser authentication page is displayed.

Typically the client address is provided by DHCP; DHCP also supplies DNS IP addresses to the client. (The DHCP server must be configured with DNS addresses that correspond to the authenticated VLANs.) See "Setting Up the DHCP Server" on page 21-29 for more information about DHCP and authentication.

For more information about authentication IP addresses, see "Configuring Authentication IP Addresses" on page 21-27.

To configure a DNS path, use the **aaa avlan dns** command. For example:

```
-> aaa avlan dns name auth.company
```

When this command is configured, a Web browser client may enter **auth.company** in the browser command line to initiate the authentication process.

To remove a DNS path from the configuration, use the **no** form of the command. For example:

```
-> no aaa avlan dns
```

The DNS path is removed from the configuration, and Web browser clients must enter the authentication IP address to initiate the authentication process.

Setting Up the DHCP Server

DHCP is a convenient way to assign IP addresses to an authentication client. DHCP will also serve DNS IP addresses to clients.

There may be one DHCP server that serves all authenticated VLANs or a DHCP server for each authenticated VLAN. The DHCP server may be located in the default VLAN, an authenticated VLAN, or both. Typically a DHCP server is located in an authenticated VLAN. Each server must be configured with IP addresses corresponding to the authenticated VLANs for which it will serve addresses.

A DHCP relay must be set up if authentication clients and the DHCP server are located in different VLANs, or if authentication clients do not belong to any VLAN. Telnet and Web browser authentication clients require IP addresses prior to authentication as well as after authenticating. The relay may be used to serve IP addresses both before and after authentication.

Note. For more information about configuring DHCP relay in general, see Chapter 17, "Configuring DHCP Relay."

Before Authentication

Normally, authentication clients cannot traffic in the default VLAN, so authentication clients do not belong to any VLAN when they connect to the switch. Even if DHCP relay is enabled, the DHCP discovery process cannot take place. To address this issue, a DHCP gateway address must be configured so that the DHCP relay "knows" which router port address to use for serving initial IP addresses. (See "Configuring a DHCP Gateway for the Relay" on page 21-31 for information about configuring the gateway address.)

Note. The switch may be set up so that authentication clients will belong to the default VLAN prior to authentication (see "Setting Up the Default VLAN for Authentication Clients" on page 21-27). If a DHCP server is located in the default VLAN, clients may obtain initial IP addresses from this server without using a relay. However, the DHCP server is typically not located in a default VLAN because it is more difficult to manage from an authenticated part of the network.

After Authentication

When the client authenticates, the client is moved into the allowed VLAN based on VLAN information sent from an authentication server (single mode authority) or based on VLAN information configured directly on the switch (multiple mode authority).

For information about authentication server authority modes, see "Configuring the Server Authority Mode" on page 21-32.

After authentication a client may be moved into a VLAN in which the client's current IP address does not correspond. This will happen if the DHCP gateway address for assigning initial IP addresses is the router port of an authenticated VLAN to which the client does not belong. (See "Configuring a DHCP Gateway for the Relay" on page 21-31.)

In this case, clients will send DHCP release/renew requests to get an address in the authenticated VLAN to which they have access; DHCP relay must be enabled so that the request can be forwarded to the appropriate VLAN.

Note. Telnet clients typically require manual configuration for IP address release/renew. Web browser clients will initiate their release/renew process automatically.

Enabling DHCP Relay for Authentication Clients

To enable DHCP relay, specify the DHCP server with the **ip helper address** command.

```
-> ip helper address 10.10.2.3
```

DHCP is automatically enabled on the switch whenever a DHCP server address is defined. For more information about using the **ip helper address** command, see Chapter 17, "Configuring DHCP Relay."

If multiple DHCP servers are used, one IP address must be configured for each server. The default VLAN DHCP gateway must also be specified so that Telnet and Web browser clients can obtain IP addresses prior to authentication. See the next section for more information.

If you want to specify that the relay only be used for packets coming in on an authenticated port, enter the **ip helper avlan only** command.

```
-> ip helper avlan only
```

When this command is specified, the switch will act as a relay for authentication DHCP packets only; non-authentication DHCP packets will not be relayed. For more information about using the **ip helper avlan only** command, see Chapter 17, "Configuring DHCP Relay."

Configuring a DHCP Gateway for the Relay

The default authenticated VLAN DHCP gateway must also be configured through the **aaa avlan default dhcp** command so that Telnet and Web browser clients can obtain IP addresses prior to authentication. This gateway is a router port in any of the authenticated VLANs in the network. It specifies the scope into which an authentication client receives an initial IP address. For example:

```
-> aaa avlan default dhcp 192.10.10.22
```

Telnet and Web browser clients will initially receive an IP address in this scope. (After authentication, these clients may require a new IP address if they do not belong to the VLAN associated with this gateway address.)

To remove a gateway address from the configuration, use the **no** form of the **aaa avlan default dhcp** command. For example:

```
-> no aaa avlan default dhcp
```

Configuring the Server Authority Mode

Authentication servers for Layer 2 authentication are configured in one of two modes: single authority or multiple authority. Single authority mode uses a single list of servers (one primary server and up to three backups) to poll with authentication requests. Multiple authority mode uses multiple lists of servers and backups, one list for each authenticated VLAN.

Note. Only one mode is valid on the switch at one time.

At least one server must be configured in either mode. Up to three backup servers total may be specified. The CLI commands required for specifying the servers are as follows:

aaa authentication vlan single-mode
aaa authentication vlan multiple-mode

Note. Each RADIUS and LDAP server may each have an additional backup host of the same type configured through the **aaa radius-server** and **aaa ldap-server** commands.

In addition, the **aaa accounting vlan** command may be used to set up an accounting server or servers to keep track of user session statistics. Setting up servers for accounting is described in "Specifying Accounting Servers" on page 21-35.

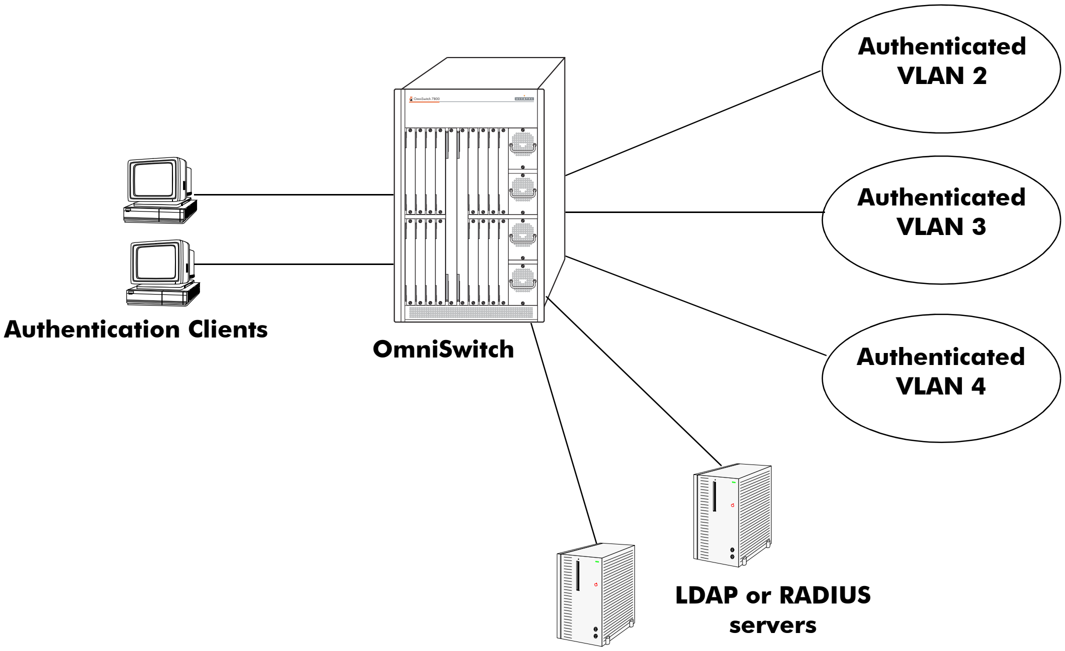
Configuring Single Mode

This mode should be used when all authenticated VLANs on the switch are using a single authentication server (with optional backups) configured with VLAN information. When this mode is configured, a client is authenticated into a particular VLAN or VLANs. (For the client to be authenticated into multiple VLANs, each VLAN must be configured for a different protocol.)

When a client first makes a connection to the switch, the agent in the switch polls the authentication server for a match with a client's user name and password. If the authentication server is down, the first backup server is polled. The switch uses the first available server to attempt to authenticate the user. (If a match is not found on that server, the authentication attempt fails. The switch does not try the next server in the list.)

If a match is found on the first available server, the authentication server sends a message to the agent in the switch that includes the VLAN IDs to which the client is allowed access. The agent then moves the MAC address of the client out of the default VLAN and into the appropriate authenticated VLAN(s).

In the illustration shown here, the Ethernet clients connect to the switch and initially belong to VLAN 1. Additional VLANs have been configured as authenticated VLANs. LDAP and RADIUS servers are configured with VLAN ID information for the clients.



Authentication Network—Single Mode

To configure authentication in single mode, use the **aaa authentication vlan** command with the **single-mode** keyword and name(s) of the relevant server and any backups. At least one server must be specified; the maximum is four servers. For example:

```
-> aaa authentication vlan single-mode ldap1 ldap2
```

In this example, authenticated VLANs are enabled on the switch in single mode. All authenticated VLANs on the switch will use **ldap1** to attempt to authenticate users. If **ldap1** becomes unavailable, the switch will use backup server **ldap2**. Both servers contain user information, including which VLANs users may be authenticated through. (The servers must have been previously set up with the **aaa ldap-server** command. For more information about setting up authentication servers, see Chapter 20, “Managing Authentication Servers.”)

To disable authenticated VLANs, use the **no** form of the command. Note that the mode does not have to be specified. For example:

```
-> no aaa authentication vlan
```

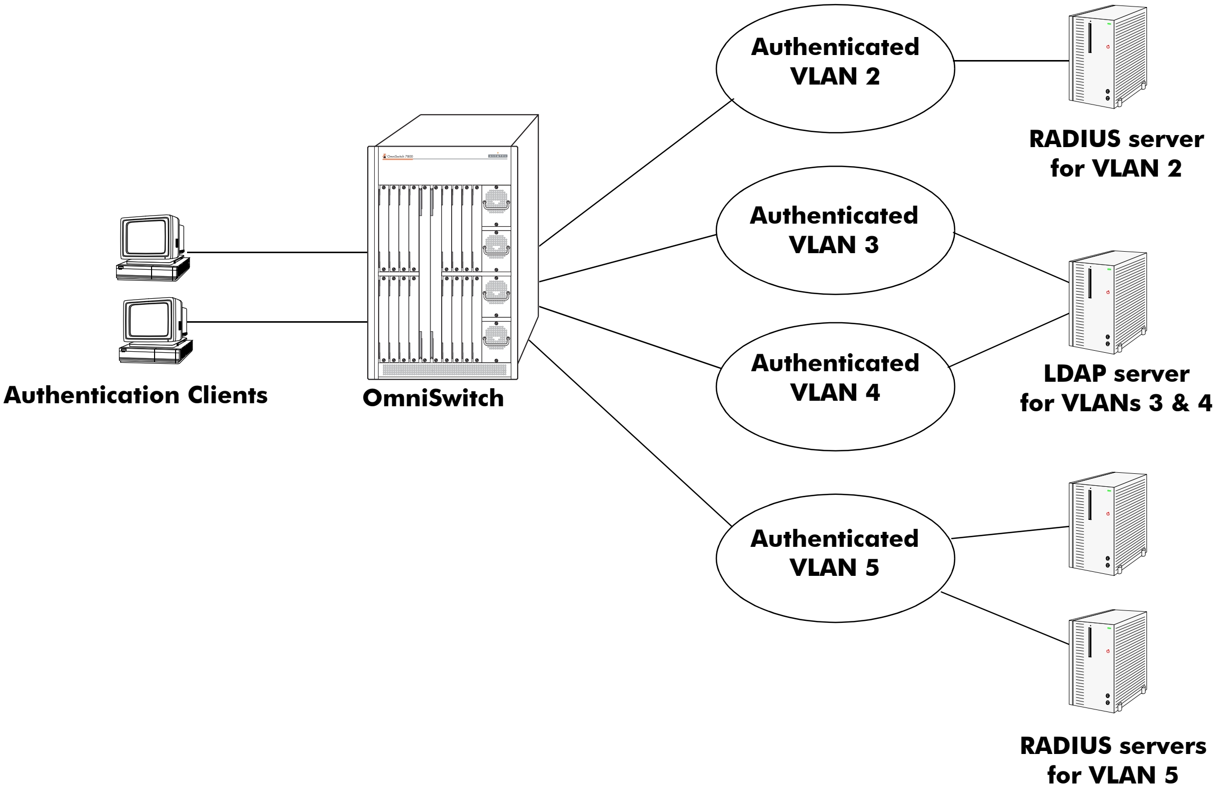
Configuring Multiple Mode

Multiple authority mode associates different servers with particular VLANs. This mode is typically used when one party is providing the network and another is providing the server.

When this mode is configured, a client is first prompted to select a VLAN. After the VLAN is selected, the client then enters a user name and password. The server configured for that particular authenticated VLAN is polled for a match. (If the server is unavailable, the switch polls the first backup server, if one is configured.) If a match is not found on the first available server, the authentication attempt fails. If a match is found, the client's MAC address is moved into that VLAN.

A server in multiple authority mode does not have to be configured with VLAN information. If the same server services more than one VLAN, the same user ID and password may be used to authenticate into one of several VLANs, depending on which VLAN the user selects at authentication. Clients are only able to authenticate into one VLAN at a time. (In single authority mode, clients can authenticate into more than one VLAN at a time if each VLAN is configured for a different protocol.)

In the illustration shown here, the clients connect to the switch and initially belong to VLAN 1. VLANs 2, 3, 4, and 5 have been configured as authenticated VLANs. A single RADIUS server is associated with VLAN 2, a primary and a backup server are associated with VLAN 5; these servers are not configured with VLAN information because each server is only serving one VLAN. However, a single LDAP server is associated with VLAN 3 and VLAN 4 and must contain VLAN information.



Authentication Network—Multiple Mode

To configure authentication in multiple mode, use the **aaa authentication vlan** command with the **multiple-mode** keyword, the relevant VLAN ID, and the names of the servers. The VLAN ID is required, and at least one server must be specified (a maximum of four servers is allowed per VLAN). For example:

```
-> aaa authentication vlan multiple-mode 2 rad1
-> aaa authentication vlan multiple-mode 3 ldap1
-> aaa authentication vlan multiple-mode 4 ldap1
-> aaa authentication vlan multiple-mode 5 ldap2 ldap3
```

To disable Authenticated VLANs in multiple mode, use the **no** form of the command and specify the relevant VLAN. Note that the mode does not have to be specified. For example:

```
-> no aaa authentication vlan 2
```

This command disables authentication on VLAN 2. VLANs 3, 4, and 5 are still enabled for authentication.

Specifying Accounting Servers

RADIUS and LDAP servers can also keep track of statistics for user authentication sessions. To specify servers to be used for accounting, use the **aaa accounting vlan** command with the relevant accounting server names. (Accounting servers are configured with the **aaa ldap-server** and **aaa radius-server** commands, which are described in Chapter 20, "Managing Authentication Servers.") Up to four accounting servers may be specified. For example:

```
-> aaa accounting vlan rad1 ldap2
```

In this example, a RADIUS server (**rad1**) is used for all accounting of authenticated VLANs; an LDAP server (**ldap2**) is specified as a backup accounting server.

If the switch is configured for multiple authority mode, the VLAN ID must be specified. In multiple mode, a different accounting server (with backups) may be specified for each VLAN. For example:

```
-> aaa accounting vlan 3 rad1 rad2 ldap1
-> aaa accounting vlan 4 ldap2 ldap3
```

In this example, **rad1** is configured an an accounting server for VLAN 3; **rad2** and **ldap1** are backups that are only used if the previous server in the list goes down. An LDAP server (**ldap2**) is configured for accounting in VLAN 4; the backup server for VLAN 4 is **ldap3**.

If an external server is not specified with the command, AVLAN user session information will be logged in the local switch log. For information about switch logging, see Chapter 30, "Using Switch Logging." In addition, the keyword **local** may be used so that logging will be done on the switch if the external server or servers become unavailable. If **local** is specified, it must be specified last in the list of servers.

In the following example, single-mode authentication is already set up on the switch, the **aaa accounting vlan** command configures a RADIUS server (**rad1**) for accounting. The local logging feature in the switch (**local**) is the backup accounting mechanism.

```
-> aaa accounting vlan rad1 local
```

Verifying the AVLAN Configuration

To verify the authenticated VLAN configuration, use the following **show** commands:

show aaa authentication vlan	Displays information about authenticated VLANs and the server configuration.
show aaa accounting vlan	Displays information about accounting servers configured for authenticated VLANs.
show avlan user	Displays MAC addresses for Authenticated VLAN users on the switch.
show aaa avlan config	Displays the current global configuration for authenticated VLANs.
show aaa avlan auth-ip	Displays the IP addresses for authenticated VLANs.

For more information about these commands, see the *OmniSwitch CLI Reference Guide*.

22 Configuring 802.1X

Physical devices attached to a LAN port on the switch through a point-to-point LAN connection may be authenticated through the switch through port-based network access control. This control is available through the IEEE 802.1X standard implemented on the switch.

In This Chapter

This chapter describes 802.1X ports used for port-based access control and how to configure them through the Command Line Interface (CLI). CLI commands are used in the configuration examples; for more details about the syntax of commands, see the *OmniSwitch CLI Reference Guide*.

This chapter provides an overview of 802.1X and includes the following information:

- "Setting Up Port-Based Network Access Control" on page 22-10.
- "Enabling 802.1X on Ports" on page 22-10.
- "Setting 802.1X Switch Parameters" on page 22-10.
- "Configuring 802.1X Port Parameters" on page 22-11.
- "Using Access Guardian Policies" on page 22-8
- "Verifying the 802.1X Port Configuration" on page 22-19.

802.1X Specifications

RFCs Supported	RFC 2284–PPP Extensible Authentication Protocol (EAP) RFC 2865–Remote Authentication Dial In User Service (RADIUS) RFC 2866–RADIUS Accounting RFC 2867–RADIUS Accounting Modifications for Tunnel Protocol Support RFC 2868–RADIUS Attributes for Tunnel Protocol Support RFC 2869–RADIUS Extensions
IEEE Standards Supported	IEEE 802.1X-2001–Standard for Port-based Network Access Control 802.1X RADIUS Usage Guidelines

802.1X Defaults

The following table lists the defaults for 802.1X port configuration through the **802.1x** command and the relevant command keywords:

Description	Keyword	Default
Port control in both directions or incoming only.	**direction {both | in}**	both
Port control authorized on the port.	**port control {force-authorized | force-unauthorized | auto}**	auto
The time during which the port will not accept an 802.1X authentication attempt.	**quiet-period**	60 seconds
The time before an EAP Request Identity will be re-transmitted.	**tx-period**	30 seconds
Number of seconds before the switch will time out an 802.1X user who is attempting to authenticate.	**supp-timeout**	30 seconds
Maximum number of times the switch will retransmit an authentication request before it times out.	**max-req**	2
Amount of time that must expire before a re-authentication attempt is made.	**re-authperiod**	3600 seconds
Whether or not the port is re-authenticated.	**no reauthentication | reauthentication**	no reauthentication

Note. By default, accounting is disabled for 802.1X authentication sessions.

Quick Steps for Configuring 802.1X

1 Configure the port as a mobile port and an 802.1X port using the following **vlan port** commands:

```
-> vlan port mobile 3/1
-> vlan port 3/1 802.1x enable
```

The port is set up automatically with 802.1X defaults. See "802.1X Defaults" on page 22-2 for information about the defaults. For more information about **vlan port** commands, See Chapter 7, "Assigning Ports to VLANs."

2 Configure the RADIUS server to be used for port authentication.

```
-> aaa radius-server rad1 host 10.10.2.1 timeout 25
```

See Chapter 20, "Managing Authentication Servers," for more information about configuring RADIUS authentication servers for 802.1X authentication.

Note. If 802.1X users will be authenticating into an authenticated VLAN, the VLAN must be configured with the **vlan authentication** command. For information about configuring VLANs with authentication, see Chapter 5, "Configuring VLANs."

3 Associate the RADIUS server (or servers) with authentication for 802.1X ports.

```
-> aaa authentication 802.1x rad1
```

4 (Optional) Associate the server (or servers) to be used for accounting (logging) 802.1X sessions. For example:

```
-> aaa accounting 802.1x rad2 ldap3 local
```

5 (Optional) Configure port-access control parameters for the 802.1X port using the **802.1x** command.

```
-> 802.1x 3/1 quiet-period 45 max-req 3
```

6 (Optional) Configure the number of times supplicant devices are polled for identification using the **802.1x supp-polling retry** command.

```
-> 802.1x 3/1 supp-polling retry 10
```

Note. Verify the 802.1X port configuration using the **show 802.1x** command:

```
-> show 802.1x 1/13
802.1x configuration for slot 1 port 13:

  direction                    = both,
  operational directions       = both,
  port-control                 = auto,
  quiet-period (seconds)       = 60,
  tx-period (seconds)          = 30,
  supp-timeout (seconds)       = 30,
  server-timeout (seconds)     = 30,
  max-req                      = 2,
  re-authperiod (seconds)      = 3600,
  reauthentication             = no
  Supplicant polling retry count = 2
```

Optional. To display the number of 802.1x users on the switch, use the **show 802.1x users** command:

```
->show 802.1x users

Slot  MAC                Port                 User
Port  Address            State                Name
-----+------------------+--------------------+--------------------------
3/1   00:60:4f:11:22:33  Connecting            user50
3/1   00:60:4f:44:55:66  Held                  user51
3/1   00:60:4f:77:88:99  Authenticated         user52
3/3   00:60:22:15:22:33  Force-authenticated   N/A
3/3   00:60:22:44:75:66  Force-authenticated   N/A
3/3   00:60:22:37:98:09  Force-authenticated   N/A
```

Optional. To display the number of non-802.1x users learned on the switch, use the **show 802.1x non-supp** command:

```
->show 802.1x non-supp

Slot  MAC               Vlan
Port  Address           Learned
-----+-----------------+----------
3/1   00:61:4f:11:22:33     2
3/1   00:61:4f:44:55:66     2
3/1   00:61:4f:77:88:99     2
3/3   00:61:22:15:22:33     5
3/3   00:61:22:44:75:66   5
```

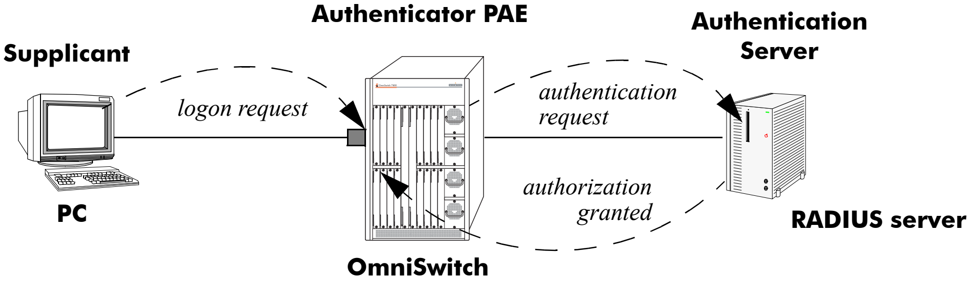
See the *OmniSwitch CLI Reference Guide* for information about the fields in this display.

802.1X Overview

The 802.1X standard defines port-based network access controls, and provides the structure for authenticating physical devices attached to a LAN. It uses the Extensible Authentication Protocol (EAP).

There are three components for 802.1X:

- **The Supplicant**—This is the device connected to the switch. The device may be connected directly to the switch or via a point-to-point LAN segment. Typically the supplicant is a PC or laptop.
- **The Authenticator Port Access Entity (PAE)**—This entity requires authentication from the supplicant. The authenticator is connected to the supplicant directly or via a point-to-point LAN segment. The OmniSwitch acts as the authenticator.
- **The Authentication Server**—This component provides the authentication service and verifies the credentials (username, password, challenge, etc.) of the supplicant. On the OmniSwitch, only RADIUS servers are currently supported for 802.1X authentication.



802.1X Components

Note. The OmniSwitch itself cannot be an 802.1X supplicant.

A device that does not use the 802.1x protocol for authentication is referred to as a *non-supplicant*. The Access Guardian feature provides configurable device classification policies to authenticate access of both supplicant and non-supplicant devices on 802.1x ports. See "Using Access Guardian Policies" on page 22-8 for more information.

Supplicant Classification

When an EAP frame or an unknown source data frame is received from a supplicant, the switch sends an EAP packet to request the supplicant's identity. The supplicant then sends the information (an EAP response), which is validated on an authentication server set up for authenticating 802.1X ports. The server determines whether additional information (a challenge, or secret) is required from the supplicant.

After the supplicant is successfully authenticated, the MAC address of the supplicant is learned in the appropriate VLAN depending on the following conditions:

- If the authentication server returns a VLAN ID, then the supplicant is assigned to that VLAN. All subsequent traffic from the supplicant is then forwarded on that VLAN.

- If the authentication server does not return a VLAN ID, then the supplicant is classified according to any device classification policies that are configured for the port. See "Using Access Guardian Policies" on page 22-8 for more information.
- If the authentication server does not return a VLAN ID and there are no user-configured device classification policies for the port, then by default Group Mobility is used to classify the supplicant. If Group Mobility is unable to classify the supplicant, then the supplicant is assigned to the default VLAN for the 802.1X port.
- If the authentication server returns a VLAN ID that does not exist or authentication fails, the supplicant is blocked.

Note that multiple supplicants can be authenticated on a given 802.1X port. Each supplicant MAC address received on the port is authenticated and learned separately. Only those that authenticate successfully are allowed on the port, as described above. Those that fail authentication are blocked on the 802.1X port.

The global configuration of this feature is controlled by the **aaa authentication 802.1x** command. This command enables 802.1X for the switch and identifies the primary and backup authentication servers. See "Setting 802.1X Switch Parameters" on page 22-10 for more information about configuring this command.

Using the **802.1x** command, an administrator may force an 802.1X port to always accept any frames on the port (therefore not requiring a device to first authenticate on the port); or an administrator may force the port to never accept any frames on the port. See "Configuring the Port Authorization" on page 22-11.

802.1X Ports and DHCP

DHCP requests on an 802.1X port are treated as any other traffic on the 802.1X port.

When the port is in an unauthorized state (which means no device has authenticated on the port), the port is blocked from receiving any traffic except 802.1X packets. This means that DHCP requests will be blocked as well.

When the port is in a forced unauthorized state (the port is manually set to unauthorized), the port is blocked from receiving all traffic, including 802.1X packets and DHCP requests.

If the port is in a forced authorized state (manually set to authorized), any traffic, including DHCP, is allowed on the port.

If the port is in an authorized state because a device has authenticated on the port, only traffic with an authenticated MAC address is allowed on the port. DHCP requests from the authenticated MAC address are allowed; any others are blocked.

Re-authentication

After a supplicant has successfully authenticated through an 802.1X port, the switch may be configured to periodically re-authenticate the supplicant (re-authentication is disabled by default). In addition, the supplicant may be manually re-authenticated (see "Re-authenticating an 802.1X Port" on page 22-12).

The re-authentication process is transparent to a user connected to the authorized port. The process is used for security and allows the authenticator (the OmniSwitch) to maintain the 802.1X connection.

Note. If the MAC address of the supplicant has aged out during the authentication session, the 802.1X software in the switch will alert the source learning software in the switch to re-learn the address.

802.1X ports may also be initialized if there a problem on the port. Initializing a port drops connectivity to the port and requires the port to be re-authenticated. See "Initializing an 802.1X Port" on page 22-13.

802.1X Accounting

802.1X authentication sessions may be logged if servers are set up for 802.1X accounting. Accounting may also be done through the local Switch Logging feature. For information about setting up accounting for 802.1X, see "Configuring Accounting for 802.1X" on page 22-13.

Compared to Authenticated VLANs

A given port cannot be both a VLAN-authenticated port and an 802.1X port. An 802.1X user, however, may be authenticated and moved into an authenticated VLAN if the RADIUS authentication server specifies a VLAN for that user and the authenticated VLAN is set up on the switch through the **vlan authentication** command. For information about configuring VLANs with authentication, see Chapter 5, "Configuring VLANs."

Both 802.1X and authenticated VLANs may use the same RADIUS authentication server. See Chapter 20, "Managing Authentication Servers," for information about using a RADIUS server for authentication.

Using Access Guardian Policies

In addition to the authentication and VLAN classification of 802.1x clients (supplicants), the Access Guardian extends this type of functionality to non-802.1x clients (non-supplicants). Access Guardian introduces configurable 802.1x device classification policies to handle both supplicant and non-supplicant access to 802.1x ports.

By default non-supplicant devices are automatically blocked on 802.1x enabled ports. In some cases, however, it is desirable to allow non-supplicant access on these ports. For example, using device classification policies, a non-supplicant device may gain access to a pre-determined VLAN. Such a VLAN might serve as a guest VLAN for non-supplicant devices that require restricted access to the switch.

Supplicant devices are initially processed using 802.1x authentication via a remote RADIUS server. If authentication is successful and returns a VLAN ID, the supplicant is assigned to that VLAN. If not, then any configured device classification policies for the port are applied to determine VLAN assignment for the supplicant. If there are no policies, then the default port behavior for 802.1x ports is in affect (see "Supplicant Classification" on page 22-5 for more information).

Policy Types

There are two type of policies: supplicant and non-supplicant. Supplicant policies use 802.1x authentication via a remote RADIUS server and provide alternative methods for classifying supplicants if the authentication process either fails or does not return a VLAN ID.

Non-supplicant policies use MAC authentication via a remote RADIUS server or can bypass authentication and only allow strict assignment to specific VLANs. MAC authentication verifies the source MAC address of a non-supplicant device via a remote RADIUS server. Similar to 802.1x authentication, the switch sends RADIUS frames to the server with the source MAC address embedded in the username and password attributes.

One supplicant and one non-supplicant policy is allowed for each 802.1x port. Configuring a new supplicant or non-supplicant policy overwrites any policies that may already exist for the port. The following types of device classification policies are available:

1. **802.1x authentication**—performs 802.1x authentication via a remote RADIUS server.
2. **MAC authentication—**performs MAC based authentication via a remote RADIUS server.
3. **Group Mobility rules**—uses Group Mobility rules to determine the VLAN assignment for a device.
4. **VLAN ID**—assigns the device to the specified VLAN.
5. **Default VLAN**—assigns a device to the default VLAN for the 802.1x port.
6. **Block—**blocks a device from accessing the 802.1x port**.**

The first policy applies only to supplicants; the second policy applies only to non-supplicants. The remaining policies apply to both supplicants and non-supplicants. Policies three through six are combined with policy one or two to provide alternative methods for classifying devices when successful authentication does not return a VLAN ID. It is also possible to configure policies three through six without also specifying policy one or two. In this case, no authentication is performed, but device classification is restricted to non-authenticated VLANs.

When multiple policies are specified when configuring a device classification policy, they form a compound policy. Compound policies that use 802.1x authentication are supplicant policies; all others are non-supplicant policies.

The order in which policies are applied to client traffic is determined by the order in which the policy was configured. For example, if a compound non-supplicant policy is configured by specifying MAC authentication, Group Mobility rules, and default VLAN, then the policies are applied in the following sequence:

1 MAC authentication is performed.

2 If authentication was successful and provided a VLAN ID, the client is assigned to that VLAN and no further policies are applied.

3 If a VLAN ID was not provided or authentication failed, then Group Mobility rules are applied.

4 If there are no Group Mobility rules that match the client traffic, then the device is learned in the default VLAN for the port.

See "Configuring Access Guardian Policies" on page 22-14 for more information about how to use and configure policies.

Setting Up Port-Based Network Access Control

For port-based network access control, 802.1X must be enabled for the switch and the switch must know which servers to use for authenticating 802.1X supplicants.

In addition, 802.1X must be enabled on each port that is connected to an 802.1X supplicant (or device). Optional parameters may be set for each 802.1X port.

The following sections describe these procedures in detail.

Setting 802.1X Switch Parameters

Use the **aaa authentication 802.1x** command to enable 802.1X for the switch and specify an authentication server (or servers) to be used for authenticating 802.1X ports. The servers must already be configured through the **aaa radius-server** command. An example of specifying authentication servers for authenticating all 802.1X ports on the switch:

```
-> aaa authentication 802.1x rad1 rad2
```

In this example, the **rad1** server will be used for authenticating 802.1X ports. If **rad1** becomes unavailable, the switch will use **rad2** for 802.1X authentication. When this command is used, 802.1X is automatically enabled for the switch.

Enabling MAC Authentication for Non-Supplicants

Use the **aaa authentication mac** command to enable MAC authentication for the switch and specify an authentication server (or servers) to be used for authenticating non-supplicants on 802.1x ports. As with enabling 802.1x authentication, the servers specified with this command must already be configured through the **aaa radius-server** command.

The following example command specifies authentication servers for authenticating non-supplicant devices on 802.1x ports:

```
-> aaa authentication mac rad1 rad2
```

Note that the same RADIUS servers can be used for 802.1x (supplicant) and MAC (non-supplicant) authentication. Using different servers for each type of authentication is allowed but not required.

For more information about using MAC authentication and classifying non-supplicant devices, see "Using Access Guardian Policies" on page 22-8 and "Configuring Access Guardian Policies" on page 22-14.

Enabling 802.1X on Ports

To enable 802.1X on a port, use the **vlan port 802.1x** command. The port must also be configured as a mobile port.

```
-> vlan port mobile 3/1
-> vlan port 3/1 802.1x enable
```

The **vlan port 802.1x** command enables 802.1X on port 1 of slot 3. The port will be set up with defaults listed in "802.1X Defaults" on page 22-2.

To disable 802.1X on a port, use the **disable** option with **vlan port 802.1x** command. For more information about **vlan port** commands, See Chapter 7, "Assigning Ports to VLANs."

Configuring 802.1X Port Parameters

By default, when 802.1X is enabled on a port, the port is configured for bidirectional control, automatic authorization, and re-authentication. In addition, there are several timeout values that are set by default as well as a maximum number of times the switch will retransmit an authentication request to the user.

All of these parameters may be configured on the same command line but are shown here configured separately for simplicity.

Configuring the Port Control Direction

To configure the port control direction, use the **802.1x** command with the **direction** keyword with **both** for bidirectional or **in** for incoming traffic only. For example:

```
-> 802.1x 3/1 direction in
```

In this example, the port control direction is set to incoming traffic only on port 1 of slot 3.

The type of port control (or authorization) is configured with the **port-control** parameter described in the next section.

Configuring the Port Authorization

Port authorization determines whether the port is open to all traffic, closed to all traffic, or open to traffic after the port is authenticated. To configure the port authorization, use the **802.1x** command with the **port-control** keyword and the **force-authorized**, **force-unauthorized**, or **auto** option.

```
-> 802.1x 3/1 port-control force-authorized
```

In this example, the port control on port 1 of slot 3 is always authorized for any traffic.

The **auto** option configures the port to be open for traffic when a device successfully completes an 802.1X authentication exchange with the switch.

Configuring 802.1X Port Timeouts

There are several timeouts that may be modified per port:

- Quiet timeout—The time during which the port will not accept an 802.1X authentication attempt after an authentication failure.
- Transmit timeout—The time before an EAP Request Identity message will be re-transmitted.
- Supplicant (or user) timeout—The time before the switch will timeout an 802.1X user who is attempting to authenticate. During the authentication attempt, the switch sends requests for authentication information (identity requests, challenge response, etc.) to the supplicant (see "Configuring the Maximum Number of Requests" on page 22-12). If the supplicant does not reply to these requests, the supplicant is timed out when the timeout expires.

To modify the quiet timeout, use the **802.1x** command with the **quiet-period** keyword. To modify the transmit timeout, use the **802.1x** command with the **tx-period** keyword. To modify the supplicant or user timeout, use the **802.1x** command with the **supp-timeout** keyword. For example:

```
-> 802.1x 3/1 quiet-period 50 tx-period 25 supp-timeout 25
```

This command changes the quiet timeout to 50 seconds; the transmit timeout to 25 seconds; and the user timeout to 25 seconds.

Note. The authentication server timeout may also be configured (with the **server-timeout** keyword) but the value is always superseded by the value set for the RADIUS server through the **aaa radius-server** command.

Configuring the Maximum Number of Requests

During the authentication process, the switch sends requests for authentication information from the supplicant. By default, the switch will send up to two requests for information. If the supplicant does not reply within the timeout value configured for the supplicant timeout, the authentication session attempt will expire. The switch will then use its quiet timeout and transmit timeout before accepting an authentication attempt or sending out an identity request.

To change the maximum number of requests sent to the supplicant during an authentication attempt, use the **max-req** keyword with the **802.1x** command. For example:

```
-> 802.1x 3/1 max-req 3
```

In this example, the maximum number of requests that will be sent is three.

Re-authenticating an 802.1X Port

An automatic reauthentication process may be enabled or disabled on any 802.1X port. The re-authentication is used to maintain the 802.1X connection (not to re-authenticate the user). The process is transparent to the 802.1X supplicant. By default, re-authentication is not enabled on the port.

To enable or disable re-authentication, use the **reauthentication** or **no reauthentication** keywords with the **802.1x** command. For example:

```
-> 802.1x 3/1 reauthentication
```

In this example, re-authentication will periodically take place on port 1 of slot 3.

The **re-authperiod** parameter may be used to configure the time that must expire before automatic re-authentication attempts. For example:

```
-> 802.1x 3/1 reauthentication re-authperiod 25
```

In this example, automatic re-authentication is enabled, and re-authentication will take place on the port every 25 seconds.

To manually re-authenticate a port, use the **802.1x re-authenticate** command. For example:

```
-> 802.1x re-authenticate 3/1
```

This command initiates a re-authentication process for port 1 on slot 3.

Initializing an 802.1X Port

An 802.1X port may be reinitialized. This is useful if there is a problem on the port. The reinitialization process drops connectivity with the supplicant and forces the supplicant to be re-authenticated. Connectivity is restored with successful re-authentication. To force an initialization, use the **802.1x initialize** command with the relevant slot/port number. For example:

```
-> 802.1x initialize 3/1
```

This command drops connectivity on port 1 of slot 3. The switch sends out a Request Identity message and restores connectivity when the port is successfully re-authenticated.

Configuring the Supplicant Polling Retry Count

To configure the number of times the switch polls an unknown device connected to an 802.1x port, use the **802.1x supp-polling retry** command. For example,

```
-> 802.1x 3/1 supp-polling retry 10
```

If after the number of polling attempts specified the device has not responded with EAP frames, then the device is identified as a non-supplicant (non-802.1x user). When this occurs, any non-supplicant device classification policies that are configured for the port are applied to the device. See "Using Access Guardian Policies" on page 22-8 for more information. If there are no such policies, then the device is blocked.

Note that the polling interval is set to 0.5 seconds between each retry and is not a configurable at this time.

Configuring Accounting for 802.1X

To log 802.1X sessions, use the **aaa accounting 802.1x** command with the desired RADIUS server names; use the keyword **local** to specify that the Switch Logging function in the switch should be used to log 802.1X sessions. RADIUS servers are configured with the **aaa radius-server** command.

```
-> aaa accounting 802.1x rad1 local
```

In this example, the RADIUS server **rad1** will be used for accounting. If **rad1** becomes unavailable, the local Switch Logging function in the switch will log 802.1X sessions. For more information about Switch Logging, see Chapter 30, "Using Switch Logging."

Configuring Access Guardian Policies

The Access Guardian provides functionality that allows the configuration of 802.1x device classification policies for supplicants (802.1x clients) and non-supplicants (non-802.1x clients). See "Using Access Guardian Policies" on page 22-8 for more information.

Configuring device classification policies is only supported on mobile, 802.1x enabled ports. In addition, the port control status for the port must allow auto authorization. See "Setting Up Port-Based Network Access Control" on page 22-10 for specific information about how to enable 802.1x functionality on a port.

As described in "Using Access Guardian Policies" on page 22-8, there are several types of policies that when combined together create either a supplicant or non-supplicant compound policy. Consider the following when configuring compound policies:

- A single policy can only appear once for a pass condition and once for a failed condition in a compound policy.
- Up to three VLAN ID policies are allowed within the same compound policy, as long as the ID number is different for each instance specified (e.g., vlan 20 vlan 30 vlan 40).
- Compound policies must terminate. The last policy must result in either blocking the device or assigning the device to the default VLAN. If a terminal policy is not specified then the block policy is used by default.
- The order in which policies are configured determines the order in which the policies are applied.

The following table provides examples of policies that were incorrectly configured and a description of the problem:

Incorrect Policy Command	**Problem**
802.1x 1/45 supplicant policy authentication pass group-mobility vlan 200 group-mobility fail block	The **group-mobility** policy is specified more than once as a pass condition.
802.1x 1/24 non-supplicant policy authentication pass vlan 20 vlan 30 vlan 40 vlan 50 fail block	More than three VLAN ID policies are specified in the same command.

Note that if no policies are configured on an 802.1x port, non-supplicants are blocked on the port and the following classification process is performed for supplicants by default:

1 802.1x authentication via remote RADIUS server is attempted.

2 If authentication fails or successful authentication returns a VLAN ID that does not exist, the device is blocked.

3 If authentication is successful and returns a VLAN ID that exists in the switch configuration, supplicant is assigned to that VLAN.

4 If authentication is successful but does not return a VLAN ID, Group Mobility rules are checked for classification.

5 If Group Mobility classification fails, the supplicant is assigned to the default VLAN ID for the 802.1x port.

Configuring Supplicant Policies

Supplicant policies are used to classify 802.1x devices connected to 802.1x-enabled switch ports when 802.1x authentication does not return a VLAN ID or authentication fails. To configure supplicant policies, use the **802.1x supplicant policy authentication** command. The following keywords are available with this command to specify one or more policies for classifying devices:

supplicant policy keywords
group mobility **vlan** **default-vlan** **block** **pass** **fail**

If no policy keywords are specified with this command, then supplicants are blocked if 802.1x authentication fails or does not return a VLAN ID. When multiple policies are specified, the policy is referred to as a compound supplicant policy. Note that the order in which parameters are configured determines the order in which they are applied.

To configure a compound supplicant policy, use the **pass** and **fail** keywords to specify which policies to apply when 802.1x authentication is successful but does not return a VLAN ID and which policies to apply when 802.1x authentication fails or returns a VLAN ID that does not exist. The **pass** keyword is implied and therefore an optional keyword. If the **fail** keyword is not used, the default action is to block the device.

Note. When a policy is specified as a policy to apply when authentication fails, device classification is restricted to assigning supplicant devices to VLANs that are *not* authenticated VLANs.

Supplicant Policy Examples

The following table provides example supplicant policy commands and a description of how the resulting policy is applied to classify supplicant devices:

Supplicant Policy Command Example	Description
802.1x 1/24 supplicant policy authentication pass group-mobility default-vlan fail vlan 43 block	If the 802.1x authentication process is successful but does not return a VLAN ID for the device, then the following occurs: **1** Group Mobility rules are applied. **2** If Group Mobility classification fails, then the device is assigned to the default VLAN for port 1/24. If the device fails 802.1x authentication, then the following occurs: **1** If VLAN 43 exists and is not an authenticated VLAN, then the device is assigned to VLAN 43. **2** If VLAN 43 does not exist or is an authenticated VLAN, then the device is blocked from accessing the switch on port 1/24.

Supplicant Policy Command Example	Description
802.1x 1/48 supplicant policy authentication group-mobility vlan 127 default-vlan	If the 802.1x authentication process is successful but does not return a VLAN ID for the device, then the following occurs: **1** Group Mobility rules are applied. **2** If Group Mobility classification fails, then the device is assigned to VLAN 127. **3** If VLAN 127 does not exist, then the device is assigned to the default VLAN for port 1/48. If the device fails 802.1x authentication, the device is blocked on port 1/48.

Configuring Non-supplicant Policies

Non-supplicant policies are used to classify non-802.1x devices connected to 802.1x-enabled switch ports. There are two types of non-supplicant policies. One type uses MAC authentication to verify the non-802.1x device. The second type does not perform any authentication and limits device assignment only to those VLANs that are not authenticated VLANs.

To configure a non-supplicant policy that will perform MAC authentication, use the **802.1x non-supplicant policy authentication** command. The following keywords are available with this command to specify one or more policies for classifying devices:

supplicant policy keywords
group mobility **vlan** **default-vlan** **block** **pass** **fail**

When multiple policies are specified, the policy is referred to as a compound non-supplicant policy. Note that the order in which parameters are configured determines the order in which they are applied.

To configure a compound non-supplicant policy, use the **pass** and **fail** keywords to specify which policies to apply when MAC authentication is successful but does not return a VLAN ID and which policies to apply when MAC authentication fails. The **pass** keyword is implied and therefore an optional keyword. If the **fail** keyword is not used, the default action is to block the device when authentication fails.

Note. When a policy is specified as a policy to apply when authentication fails, device classification is restricted to assigning non-supplicant devices to VLANs that are *not* authenticated VLANs.

To configure a non-supplicant policy that will *not* perform MAC authentication, use the **802.1x non-supplicant policy** command. The following keywords are available with this command to specify one or more policies for classifying devices:

supplicant policy keywords
group mobility **vlan** **default-vlan** **block**

Note that this type of policy does not use 802.1x or MAC authentication. As a result, all of the available policy keywords restrict the assignment of the non-supplicant device to only those VLANs that are non-authenticated VLANs. The **pass** and **fail** keywords are not used when configuring this type of policy.

Non-supplicant Policy Examples

The following table provides example non-supplicant policy commands and a description of how the resulting policy is applied to classify supplicant devices:

Supplicant Policy Command Example	Description
802.1x 1/24 non-supplicant policy authentication pass group-mobility default-vlan fail vlan 10 block	If the MAC authentication process is successful but does not return a VLAN ID for the device, then the following occurs: **1** Group Mobility rules are applied. **2** If Group Mobility classification fails, then the device is assigned to the default VLAN for port 1/24. If the device fails MAC authentication, then the following occurs: **1** If VLAN 10 exists and is not an authenticated VLAN, the device is assigned to VLAN 10. **2** If VLAN 10 does not exist or is an authenticated VLAN, the device is blocked from accessing the switch on port 1/24.
802.1x 1/48 non-supplicant policy authentication vlan 10 default-vlan	If the MAC authentication process is successful but does not return a VLAN ID for the device, then the following occurs: **1** The device is assigned to VLAN 10. **2** If VLAN 10 does not exist, then the device is assigned to the default VLAN for port 1/48. If the device fails MAC authentication, the device is blocked from accessing the switch on port 1/48.
802.1x 2/1 non-supplicant policy authentication fail vlan 100 default-vlan	If MAC authentication does not return a VLAN ID, the device is blocked from accessing the switch on port 2/1. If the device fails MAC authentication, then the following occurs: **1** If VLAN 100 exists and is not an authenticated VLAN, the device is assigned to VLAN 100. **2** If VLAN 100 does not exist or is an authenticated VLAN, the device is assigned to the default VLAN for port 2/1. **3** If the default VLAN for port 2/1 is an authenticated VLAN, then the device is blocked from accessing the switch on port 2/1.

Supplicant Policy Command Example	Description
802.1x 2/10 non-supplicant policy authentication pass vlan 10 block fail group-mobility default-vlan	If the MAC authentication process is successful but does not return a VLAN ID for the device, then the following occurs: **1** The device is assigned to VLAN 10. **2** If VLAN 10 does not exist, then the device is blocked from accessing the switch on port 2/10. If the device fails MAC authentication, then the following occurs: **1** Group Mobility rules are applied. **2** If Group Mobility classification fails, then the device is assigned to the default VLAN for port 2/10. **3** If the default VLAN for port 2/10 is an authenticated VLAN, then the device is blocked from accessing the switch on port 2/10.
802.1x 3/1 non-supplicant policy authentication pass vlan 10 block fail group-mobility vlan 43 default-vlan	If the MAC authentication process is successful but does not return a VLAN ID for the device, then the following occurs: **1** The device is assigned to VLAN 10. **2** If VLAN 10 does not exist, then the device is blocked from accessing the switch on port 3/1. If the device fails MAC authentication, then the following occurs: **1** Group Mobility rules are applied. **2** If Group Mobility classification fails, then the device is assigned to VLAN 43. **3** If VLAN 43 does not exist or is an authenticated VLAN, then the device is assigned to the default VLAN for port 3/1. **4** If the default VLAN for port 3/1 is an authenticated VLAN, then the device is blocked from accessing the switch on port 3/1.
802.1x 3/10 non-supplicant policy vlan 43 block	No authentication process is performed.but the following classification still occurs: **1** If VLAN 43 exists and is not an authenticated VLAN, then the device is assigned to VLAN 43. **2** If VLAN 43 does not exist or is an authenticated VLAN, then the device is blocked from accessing the switch on port 3/10.

Verifying the 802.1X Port Configuration

A summary of the **show** commands used for verifying the 802.1X port configuration is given here:

show 802.1x Displays information about ports configured for 802.1X.

show 802.1x users Displays a list of all users (supplicants) for one or more 802.1X ports.

show 802.1x non-supp Displays a list of all non-802.1x users (non-supplicants) learned on one or more 802.1x ports.

show 802.1x statistics Displays statistics about 802.1X ports.

show 802.1x device classification policies Displays Access Guardian device classification policies configured for 802.1x ports.

show aaa authentication 802.1x Displays information about the global 802.1X configuration on the switch.

show aaa accounting 802.1x Displays information about accounting servers configured for 802.1X port-based network access control.

show aaa authentication mac Displays a list of RADIUS servers configured for MAC based authentication.

For more information about the displays that result from these commands, see the *OmniSwitch CLI Reference Guide*.

23 Managing Policy Servers

Quality of Service (QoS) policies that are configured through Alcatel's PolicyView network management application are stored on a Lightweight Directory Access Protocol (LDAP) server. PolicyView is an OmniVista application that runs on an attached workstation.

In This Chapter

This chapter describes how LDAP directory servers are used with the switch for policy management. There is no required configuration on the switch. When policies are created on the directory server through PolicyView, the PolicyView application automatically configures the switch to communicate with the server. This chapter includes information about modifying configuration parameters through the Command Line Interface (CLI) if manual reconfiguration is necessary. For more details about the syntax of commands, see the *OmniSwitch CLI Reference Guide*.

Throughout this chapter the term *policy server* is used to refer to LDAP directory servers used to store policies. Procedures described in this chapter include:

- "Installing the LDAP Policy Server" on page 23-3
- "Modifying Policy Servers" on page 23-4
- "Verifying the Policy Server Configuration" on page 23-7

Policy Server Specifications

The following tables lists important information about LDAP policy servers:

LDAP Policy Servers RFCs Supported	RFC 2251–Lightweight Directory Access Protocol (v3) RFC 3060–Policy Core Information Model – Version 1 Specification
Maximum number of policy servers (supported on the switch)	4
Maximum number of policy servers (supported by PolicyView)	1

Policy Server Defaults

Defaults for the **policy server** command are as follows:

Description	**Keyword**	**Default**
The port number for the server	**port**	389 (SSL disabled) 636 (SSL enabled)
Priority value assigned to a server, used to determine search order	**preference**	0 (lowest)
Whether a Secure Socket Layer is configured for the server	**ssl | no ssl**	no ssl

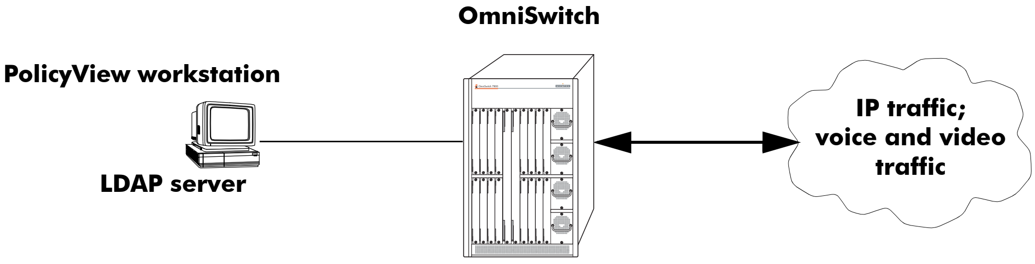
Policy Server Overview

The Lightweight Directory Access Protocol (LDAP) is a standard directory server protocol. The LDAP policy server client in the switch is based on RFC 2251. Currently, only LDAP servers are supported for policy management.

When the policy server is connected to the switch, the switch is automatically configured to communicate with the server to download and manage policies created by the PolicyView application. There is no required user configuration. (Note that the LDAP policy server is automatically installed when the PolicyView application is installed.)

Note. The switch has separate mechanisms for managing QoS policies stored on an LDAP server and QoS policies configured directly on the switch. For more information about creating policies directly on the switch, see Chapter 24, “Configuring QoS.”

Information about installing the LDAP policy server is included in this chapter. Consult the server manufacturer’s documentation for detailed information about configuring the server.



Policy Server Setup

Installing the LDAP Policy Server

Currently Netscape Directory Server 4.15 is supported. The server software is bundled with the PolicyView NMS application.

1 Install the directory server software on the server.

2 Install the Java Runtime Environment on the server.

See your server documentation for additional details on setting up the server.

See the next sections of this chapter for information about modifying policy server parameters or viewing information about policy servers.

Modifying Policy Servers

Policy servers are automatically configured when the server is installed; however, policy server parameters may be modified if necessary.

Note. SSL configuration must be done manually through the **policy server** command.

Modifying LDAP Policy Server Parameters

Use the **policy server** command to modify parameters for an LDAP policy server.

Keywords for the command are listed here:

Policy server keywords	
port **admin** **preference** **user**	**password** **searchbase** **ssl**

For information about policy server parameter defaults, see "Policy Server Defaults" on page 23-2.

Disabling the Policy Server From Downloading Policies

Policy servers may be prevented from downloading policies to the switch. By default, policy servers are enabled to download policies.

To disable a server, use the **policy server** command with the **admin** keyword and **down** option.

```
-> policy server 10.10.2.3 admin down
```

In this example, an LDAP server with an IP address of 10.10.2.3 will not be used to download policies. Any policies already downloaded to the switch are not affected by disabling the server.

To re-enable the server, specify **up**.

```
-> policy server 10.10.2.3 admin up
```

The server is now available for downloading policies.

To delete a policy server from the configuration, use the **no** form of the command with the relevant IP address:

```
-> no policy server 10.10.2.3
```

If the policy server is not created on the default port, the **no** form of the command must include the port number. For example:

```
-> no policy server 10.10.2.4 5000
```

Modifying the Port Number

To modify the port, enter the **policy server** command with the **port** keyword and the relevant port number.

```
-> policy server 10.10.2.3 port 636
```

Note that the port number must match the port number configured on the policy server.

Also, if the port number is modified, any existing entry for that policy server is not removed. Another entry is simply added to the policy server table.

Note. If you enable SSL, the port number is automatically set to 636. (This does not create another entry in the port table.)

For example, if you configure a policy server with port 389 (the default), and then configure another policy server with the same IP address but port number 5000, two entries will display on the **show policy server** screen.

```
-> policy server 10.10.2.3
-> policy server 10.10.2.3 port number 5000
-> show policy server

Server IP Address     port   enabled    status   primary
------+--------------+-----+---------+--------+--------
     1  10.10.2.3      389    Yes        Up       X
     2  10.10.2.3      5000   No         Down     -
```

To remove an entry, use the **no** form of the **policy server** command. For example:

```
-> no policy server 10.10.2.3 port number 389
```

The first entry is removed from the policy server table.

Modifying the Policy Server Username and Password

A user name and password may be specified so that only specific users can access the policy server.

```
-> policy server 10.10.2.3 user kandinsky password blue
```

If this command is entered, a user with a username of **kandinsky** and a password of **blue** will be able to access the LDAP server to modify parameters on the server itself.

Modifying the Searchbase

The searchbase name is “o=alcatel.com” by default. To modify the searchbase name, enter the **policy server** command with the **searchbase** keyword. For example:

```
-> policy server 10.10.2.3 searchbase "ou=qo,o=company,c=us"
```

Note that the searchbase path must be a valid path in the server directory structure.

Configuring a Secure Socket Layer for a Policy Server

A Secure Socket Layer (SSL) may be configured between the policy server and the switch. If SSL is enabled, the PolicyView application can no longer write policies to the LDAP directory server.

By default, SSL is disabled. To enable SSL, use the **policy server** command with the **ssl** option. For example:

```
-> policy server 10.10.2.3 ssl
```

SSL is now enabled between the specified server and the switch. Note that the port number must be configured if SSL is enabled. By default the port number is 389. Typically the port number for SSL is 636; however, the port number should be configured with whatever port number is set on the server. For information about configuring the port number, see "Modifying the Port Number" on page 23-5.

To disable SSL, use **no ssl** with the command:

```
-> policy server 10.10.2.3 no ssl
```

SSL is disabled for the 10.10.2.3 policy server. No additional policies may be saved to the directory server from the PolicyView application.

Loading Policies From an LDAP Server

To download policies (or rules) from an LDAP server to the switch, use the **policy server load** command. Before a server can download policies, it must also be set up and operational (able to bind).

To download policies from the server, enter the following:

```
-> policy server load
```

Use the **show policy server long** command to display the last load time. For example:

```
-> show policy server long
LDAP server 0
  IP address                : 10.10.2.3,
  TCP port                  : 16652,
  Enabled                   : Yes,
  Operational Status        : Down,
  Preference                : 99,
  Authentication            : password,
  SSL                       : Disabled,
  login DN                  : cn=DirMgr
  searchbase                : o=company
  Last load time            : 02/14/02 16:38:18
```

Removing LDAP Policies From the Switch

To flush LDAP policies from the switch, use the **policy server flush** command. Note that any policies configured directly on the switch through the CLI *are not affected* by this command.

```
-> policy server flush
```

Interaction With CLI Policies

Policies configured via PolicyView can only be modified through PolicyView. They cannot be modified through the CLI. Any policy management done through the CLI only affects policies configured through the CLI. For example, the **qos flush** command only removes CLI policies; LDAP policies are not affected.

Also, the **policy server flush** command removes only LDAP policies; CLI policies are not affected.

Note. If polices are applied from PolicyView or vice versa, it will activate all current configuration.

For more information about configuring policies through the CLI, see Chapter 24, "Configuring QoS."

Verifying the Policy Server Configuration

To display information about authentication and policy servers, use the following commands:

show policy server	Displays information about servers from which policies may be downloaded to the switch.
show policy server long	Displays detailed information about an LDAP policy server.
show policy server statistics	Displays statistics about policy directory servers.
show policy server rules	Displays the names of policies originating on a directory server that have been downloaded to the switch.
show policy server events	Displays any events related to a directory server.

24 Configuring QoS

Alcatel's QoS software provides a way to manipulate flows coming through the switch based on user-configured policies. The flow manipulation (generally referred to as *Quality of Service* or *QoS*) may be as simple as allowing/denying traffic, or as complicated as remapping 802.1p bits from a Layer 2 network to ToS values in a Layer 3 network.

While policies may be used in many different types of network scenarios, there are several typical types discussed in this chapter:

- **Basic QoS**—includes traffic prioritization and bandwidth shaping
- **ICMP policies**—includes filtering, prioritizing, and/or rate limiting ICMP traffic for security
- **802.1p/ToS/DSCP**—includes policies for marking and mapping
- **Network Address Translation (NAT)**—includes static and dynamic address translation
- **Policy Based Routing (PBR)**—includes policies for redirecting routed traffic
- **Server Load Balancing (SLB)**
  Setting up QoS is only part of the required configuration for SLB. For more details, see Chapter 27, "Configuring Server Load Balancing."
- **Access Control Lists (ACLs)**
  ACLs are a specific type of QoS policy used for Layer 2, Layer 3/4, and multicast filtering. Since filtering is used in many different network situations, ACLs are described in a separate chapter (see Chapter 25, "Configuring ACLs").

In This Chapter

This chapter describes QoS in general and how policies are used on the switch. It provides information about configuring QoS through the Command Line Interface (CLI). CLI commands are used in the configuration examples; for more details about the syntax of commands, see the *OmniSwitch CLI Reference Guide*.

Configuration procedures described in this chapter include:

- Setting up global QoS parameters (see page 24-15)
- Setting up policy components, such as policy conditions and actions (see page 24-37)
- Configuring specific types of policies (see page 24-65)

Note. Policies may also be configured through the PolicyView NMS application and stored on an attached LDAP server. For more information about managing LDAP policies, see Chapter 23, "Managing Policy Servers."

QoS Specifications

Maximum number of policy rules	2048
Maximum number of policy conditions	2048
Maximum number of policy actions	2048
Maximum number of policy services	256
Maximum number of groups (network, MAC, service, port)	1024
Maximum number of group entries	512 per group
Maximum number of flows	64000
CLI Command Prefix Recognition	Some QoS commands support prefix recognition. See the “Using the CLI” chapter in the *OmniSwitch 7700/7800/8800 Switch Management Guide* for more information.

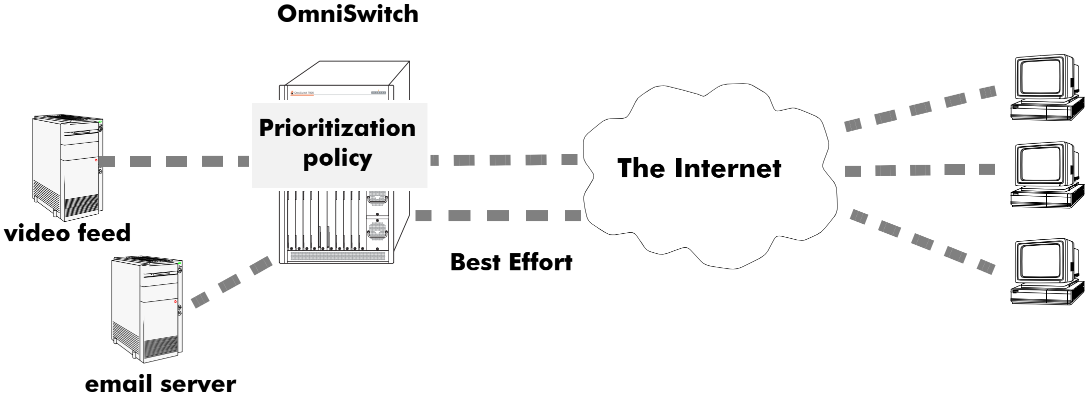
QoS General Overview

Quality of Service (QoS) refers to transmission quality and available service that is measured and sometimes guaranteed in advance for a particular type of traffic in a network. QoS lends itself to circuit-switched networks like ATM, which bundle traffic into cells of the same length and transmit the traffic over predefined virtual paths. In contrast, IP and other packet-switched networks operate on the concept of shared resources and *best effort* routing, using bandwidth as needed and reassembling packets at their destinations. Applying QoS to packet-switched networks requires different mechanisms than those used in circuit-switched networks.

QoS is often defined as a way to manage bandwidth. Another way to handle different types of flows and increased bandwidth requirements is to add more bandwidth. But bandwidth can be expensive, particularly at the WAN connection. If LAN links that connect to the WAN are not given more bandwidth, bottlenecks may still occur. Also, adding enough bandwidth to compensate for peak load periods will mean that at times some bandwidth will be unused. In addition, adding bandwidth does not guarantee any kind of control over network resources.

Using QoS, a network administrator can gain more control over networks where different types of traffic (or flows) are in use or where network congestion is high. Preferential treatment may be given to individual flows as required. Voice over IP (VoIP) traffic or mission-critical data may be marked as priority traffic and/or given more bandwidth on the link. QoS can also prevent large flows, such as a video stream, from consuming all the link's bandwidth. Using QoS, a network administrator can decide which traffic needs preferential treatment, and which traffic can be adequately served with best effort.

QoS is implemented on the switch through the use of user-defined policies. The following simplified illustration shows how video traffic may receive priority over email traffic.



Sample QoS Setup

QoS Policy Overview

A policy (or a *policy rule*) is made up of a condition and an action. The condition specifies parameters that the switch will examine in incoming flows, such as destination address or Type of Service (ToS) bits. The action specifies what the switch will do with a flow that matches the condition; for example, it may queue the flow with a higher priority, or reset the ToS bits.

Policies may be created directly on the switch through the CLI or WebView. Or policies may be created on an external LDAP server via the PolicyView application. The switch makes a distinction between policies created on the switch and policies created on an LDAP server.

Note. Polices may be only be modified using the same source used to create them. Policies configured through PolicyView may only be edited through PolicyView. Policies created directly on the switch through the CLI or WebView may only be edited on the switch. Policies may be created through the CLI or WebView, however, to override policies created in PolicyView. And vice versa.

This chapter discusses policy configuration using the CLI. For information about using WebView to configure the switch, see the *OmniSwitch 7700/7800/8800 Switch Management Guide*. For information about configuring policies through PolicyView, see the PolicyView online help.

How Policies Are Used

When a flow comes into the switch, the QoS software in the switch checks to see if there are any policies with conditions that match the flow.

- ***If there are no policies that match the flow***, the flow is accepted or denied based on the global disposition set for the switch. By default, the disposition is **accept**. Use the **qos default bridged disposition**, **qos default routed disposition**, or **qos default multicast disposition** command to change the disposition. If the flow is accepted, it is placed in a default queue on the output port.
- ***If there is more than one policy that matches the flow***, the switch uses the policy with the highest precedence. For more information about policy precedence, see "Rule Precedence" on page 24-42.
- ***Flows must also match all parameters configured in a policy condition.*** A policy condition must have at least one classification parameter.

Once the flow is classified and matched to a policy, the switch enforces the policy by mapping each packet of the flow to the appropriate queue and scheduling it on the output port. The queue may be a QoS queue or a default queue, depending on the information indicated in the policy action. For more information about queues, see "QoS Ports and Queues" on page 24-24.

Valid Policies

The switch does not allow you to create invalid condition/action combinations; if you enter an invalid combination, an error message will display.

A list of valid condition and condition/action combinations is given in "Condition Combinations" on page 24-6 and "Condition/Action Combinations" on page 24-8.

It is possible to configure a valid QoS rule that is active on the switch, however the switch is not able to enforce the rule because some other switch function (for example, routing) is disabled. See the condition and condition/action combinations tables for more information about valid combinations ("Condition Combinations" on page 24-6 and "Condition/Action Combinations" on page 24-8).

Interaction With Other Features

QoS policies may be an integral part of configuring other switch features, such as Server Load Balancing or Link Aggregation. In addition, QoS settings may affect other features in the switch; or QoS settings may require that other switch features be configured in a particular way.

A summary of related features is given here:

- **Server Load Balacing (SLB)**—SLB is used to balance traffic destined for a single IP address. SLB and QoS are described briefly in "Server Load Balancing" on page 24-73. For more details, see Chapter 27, "Configuring Server Load Balancing."
- **Dynamic Link Aggregates**—Policies may be used to prioritize dynamic link aggregation groups. For details, see Chapter 12, "Configuring Dynamic Link Aggregation."
- **802.1Q**—Tagged ports are always trusted, regardless of QoS settings. For information about configuring ports with 802.1Q, see Chapter 10, "Configuring 802.1Q."
- **Mobile Ports**—Mobile ports are always trusted, regardless of QoS settings. For information about setting up mobile ports, see Chapter 7, "Assigning Ports to VLANs."
- **IP Routing**—Some policy rules require that IP routing be enabled. See Chapter 13, "Configuring IP."
- **LDAP Policy Management**—Policies may also be configured through the PolicyView application and stored on an attached LDAP server. LDAP policies may only be modified through PolicyView. For information about setting up a policy server and managing LDAP policies, see Chapter 23, "Managing Policy Servers."

Condition Combinations

The CLI prevents you from configuring invalid condition combinations that are never allowed; however, it does allow you to create combinations that are supported in some scenario. For example, you might configure destination slot/port and destination interface type for the same condition. This is a valid combination, but will only be used to classify bridged traffic.

Note the following:

- Layer 2 and Layer 3/4 conditions cannot be combined.
- Type of Service (ToS) and (Differentiated Services Code Point) DSCP conditions cannot be combined.
- Individual items and their corresponding groups cannot be combined in the same condition. For example, a source IP address cannot be included in a condition with a source IP network group.

Use the policy condition combinations table as a guide when configuring policy conditions.

How to Use the Condition Combination Table. Each row represents items that may be combined; any cavaets are listed in the Notes column. For example, you can combine destination VLAN and destination interface type for bridged traffic only.

For information about combining conditions with actions, see "Condition/Action Combinations" on page 24-8.

Policy Condition Combinations

Condition Combinations	Supported When?	Notes
source MAC address *or* MAC group source VLAN source slot/port *or* port group source interface type	bridging only	—
source IP address *or* network group destination IP address *or* network group source TCP/UDP port destination TCP/UDP port ToS *or* DSCP IP protocol	routing; or routing/bridging when **qos classifyl3 bridged** is enabled	*IP protocol may only be specified if source and/or destination IP port is specified generally with the* ***source ip port*** *and/or* ***destination ip port*** *keywords.*
source IP address *or* network group destination interface type	any routing; or routing/bridging when **qos classifyl3 bridged** is enabled	—
source IP address *or* network group destination slot/port or port group	any routing; or routing/bridging when **qos classifyl3 bridged** is enabled	—
destination MAC address *or* MAC group destination VLAN 802.1p destination slot/port *or* port group destination interface type	bridging only	—
destination IP address *or* network group destination slot/port *or* port group	routing only	*Destination slot/port and destination interface type may only be combined when bridging. See previous row.*
destination IP address *or* network group destination interface type	routing only	—

Policy Condition Combinations

Condition Combinations	**Supported When?**	**Notes**
IP protocol destination slot/port *or* port group	routing only	*IP protocol may only be specified if source and/or destination IP port is specified generally with the* ***source ip port*** *and/or* ***destination ip port*** *keywords.; those combinations are supported when Layer 3 traffic is classified as bridged.*
IP protocol destination interface type	routing only	*IP protocol is usually combined with source or destination TCP/UDP port; those combinations are supported when Layer 3 traffic is classified as bridged.*
multicast IP address *or* network group destination IP address *or* network group destination VLAN	multicast rules only	—
multicast IP address *or* network group destination MAC address *or* MAC group	multicast rules only	—
multicast IP address *or* network group destination slot/port *or* port group	multicast rules only	—
multicast IP address *or* network group destination interface type	multicast rules only	—
multicast IP address *or* network group destination slot/port *or* port group destination interface type	multicast rules only	*Destination slot/port and destination interface type may only be combined when bridging.*

Condition/Action Combinations

Conditions and actions are combined in policy rules. The CLI prevents you from configuring invalid condition/action combinations that are never allowed; however, it does allow you to create combinations that are supported in some scenario. For example, a destination MAC address condition may be combined with an action specifying priority for flows that are bridged only.

Note the following:

- Layer 2 and Layer 3/4 conditions cannot be combined.
- Type of Service (ToS) and Differentiated Services Code Point (DSCP) cannot be combined in a condition. 802.1p, ToS, and DSCP cannot be combined in an action.
- Individual items and their correponding groups cannot be combined in the same condition. For example, a source IP address cannot be included in a condition with a source IP network group.
- Policy actions for Network Address Translation (rewrite IP addresses) cannot be combined with any other action.
- Policy actions for Server Load Balancing (load balance group) cannot be combined with any other action.
- If you combine priority with 802.1p, DSCP, or ToS in an action, the priority value will be used to prioritize the flow.
- Source or destination TCP/UDP port and IP protocol may be combined in the condition if bridged traffic is classified as Layer 3.

Use the policy condition/action combinations table as a guide when creating policy rules.

How to Use the Condition/Action Combination Table. Each row represents a policy condition or conditions that may be combined with a policy action or actions in the same row. *Note that the table does not represent every possible condition combination.* Use the Policy Condition Combinations table on page 24-6 to cross-reference supported policy conditions.

For more information about policy condition combinations, see "Condition Combinations" on page 24-6.

Policy Condition/Action Combinations

Conditions	Actions	Supported When?
source IP address *or* network group	ToS *or* DSCP	routing only
source IP address *or* network group	source rewrite IP *or* rewrite group	routing only
source IP address *or* network group	destination rewrite IP *or* rewrite group	routing only
source IP address *or* network group	load balance group	routing only
source MAC address *or* MAC group	disposition	always
source VLAN	disposition	always
source slot/port *or* port group	disposition	always
source interface type	disposition	always
source TCP/UDP port	ToS *or* DSCP	routing only

Policy Condition/Action Combinations

Conditions	Actions	Supported When?
source TCP/UDP port	source rewrite IP *or* rewrite group	routing only
source TCP/UDP port	destination rewrite IP *or* rewrite group	routing only
source IP address *or* network group destination IP address *or* network group source TCP/UDP port destination TCP/UDP port IP protocol	disposition minimum bandwidth maximum bandwidth maximum buffers priority shared 802.1p	any routing; or routing/bridging when **qos classifyl3 bridged** is enabled
source IP address *or* network group destination IP address *or* network group source TCP/UDP port IP protocol ToS or DSCP	minimum bandwidth maximum bandwidth priority shared	any routing; or routing/bridging when **qos classifyl3 bridged** is enabled
destination IP address *or* network group	ToS *or* DSCP	routing only
destination IP address *or* network group	source rewrite IP *or* rewrite group	routing only
destination IP address *or* network group	destination rewrite IP *or* rewrite group	routing only
destination IP address *or* network group	load balance group	routing only
destination MAC address *or* MAC group destination VLAN	disposition minimum bandwidth maximum bandwidth maximum buffers priority shared 802.1p *or* ToS *or* DSCP	bridging only
destination slot/port *or* port group	disposition minimum bandwidth maximum bandwidth maximum buffers priority shared 802.1p *or* ToS *or* DSCP	always
destination slot/port *or* port group	source rewrite IP *or* rewrite group	routing only
destination interface type	disposition minimum bandwidth maximum bandwidth maximum buffers priority shared 802.1p *or* ToS *or* DSCP	always
destination interface type	source rewrite IP *or* rewrite group	routing only
destination interface type	destination rewrite IP *or* rewrite group	routing only
destination TCP/UDP port	ToS *or* DSCP	routing only
destination TCP/UDP port	source rewrite IP *or* rewrite group	routing only
destination TCP/UDP port	destination rewrite IP *or* rewrite group	routing only
IP protocol	ToS *or* DSCP	routing only

Policy Condition/Action Combinations

Conditions	Actions	Supported When?
IP protocol	source rewrite IP *or* rewrite group	routing only
IP protocol	destination rewrite IP *or* rewrite group	routing only
802.1p	priority shared	bridging only
802.1p *or* ToS *or* DSCP	802.1p *or* ToS *or* DSCP	always
multicast IP address *or* network group	disposition	always
source IP address *or* network group destination IP address *or* network group source TCP/UDP port destination TCP/UDP port IP protocol	gateway	routing only
source port group (*built-in only*)	permanent gateway	routing only
source IP address *or* network group destination IP address *or* network group source TCP/UDP port destination TCP/UDP port IP protocol	alternate gateway	routing only
source port group (*built-in only*)	permanent gateway	routing only

QoS Defaults

The following tables list the defaults for global QoS parameters, individual port settings, policy rules, and default policy rules.

Global QoS Defaults

Use the **qos reset** command is to reset global values to their defaults.

Description	Command	Default
QoS enabled or disabled	**qos**	enabled
Whether ports are globally trusted or untrusted	**qos trust ports**	802.1Q-tagged ports and mobile ports are always trusted; any other port is untrusted
Number of default queues	**qos default queues**	4 queues
Statistics interval	**qos stats interval**	60 seconds
Global bridged disposition	**qos default bridged disposition**	accept
Global routed disposition	**qos default routed disposition**	accept
Global multicast disposition	**qos default multicast disposition**	accept
Flow timeout	**qos flow timeout**	300 seconds
Reflexive flow timeout	**qos reflexive timeout**	120 seconds
Fragment timeout	**qos fragment timeout**	10 seconds
NAT timeout	**qos nat timeout**	3600 seconds
Level of log detail	**qos log level**	6
Number of lines in QoS log	**qos log lines**	256
Whether log messages are sent to the console	**qos log console**	no
Whether log messages are available to OmniVista applications	**qos forward log**	no
Type of messages logged	**debug qos**	info
Whether fragments are classified	**qos classify fragments**	no
Whether bridged traffic may be classified with Layer 3 conditions	**qos classifyl3 bridged**	no

QoS Port Defaults

Use the **qos port reset** command to reset port settings to the defaults.

Description	Command/keyword	Default
Number of default queues	**qos port default queues**	4 (*currently not configurable*)
Whether the port is trusted or untrusted	**qos port trusted**	802.1Q and mobile ports are always trusted; other ports are untrusted
The 802.1p setting to map on ingress flows to an untrusted port	**qos port default 802.1p**	0
The ToS/DSCP setting to map on ingress flows to an untrusted port	**qos port default dscp**	0
Maximum reserve bandwidth	**qos port maximum reserve bandwidth**	port bandwidth; *currently not supported.*
Maximum signalled bandwidth (via RSVP)	**qos port maximum signal bandwidth**	port bandwidth; *currently not supported.*
Maximum default depth	**qos port maximum default depth**	2048 bytes; *currently not supported.*
Maximum default buffers	**qos port maximum default buffers**	64 buffers (ENI interfaces) 192 buffers (GNI interfaces); *currently not supported.*

Policy Rule Defaults

The following are defaults for the **policy rule** command:

Description	Keyword	Default
Policy rule enabled or disabled	**enable | disable**	enabled
Determines the order in which rules are searched	**precedence**	0
Whether or not the rule is reflexive	**reflexive**	Rules are not reflexive.
Whether the rule is saved to flash immediately	**save**	Save option is enabled.

Policy Action Defaults

The following are defaults for the **policy action** command:

Description	Keyword	Default
Whether the flow matching the rule should be accepted or denied	**disposition**	accept

Note that in the current software release, the **deny** and **drop** options produce the same effect that is, the traffic is silently dropped.

Note. There are no defaults for the **policy condition** command.

Default (Built-in) Policies

The switch includes some built-in policies, or default policies, for particular traffic types or situations where traffic does not match any policies. In all cases, the switch accepts the traffic and places it into default queues.

- *Fragments*—Fragments with unknown source or destination TCP/UDP ports are accepted on the switch, unless the **qos classify fragments** command is entered. See "Fragment Classification" on page 24-21 for more information about this command.
- *Other traffic*—Any traffic that does not match a policy is accepted or denied based on the global disposition setting on the switch. The global disposition is by default **accept**. Use the **qos default bridged disposition, qos default routed disposition,** and **qos default multicast disposition** commands to change the disposition as described in "Creating Policy Conditions" on page 24-39 and "Setting the Global Default Dispositions" on page 24-15.
- *The* **switch** *network group*—The switch has a default network group, called **switch**, that includes all IP addresses configured for the switch itself. This default network group may be used in policies. See "Creating Network Groups" on page 24-50 for more information about network groups.
- *Policy Port Groups*—The switch has built-in policy port groups for each slot. The groups are called **Slot01**, **Slot02**, etc. Use the **show policy port group** command to view the built-in groups.

QoS Configuration Overview

QoS configuration involves the following general steps:

1 Configuring Global Parameters. In addition to enabling/disabling QoS, global configuration includes settings such as global port parameters, default disposition for flows, and various timeouts. The type of parameters you might want to configure globally will depend on the types of policies you will be configuring. For example, if you want to set up policies for 802.1p or ToS/DSCP traffic, you may want to configure all ports as trusted ports.

Typically, you will not need to change any of the global defaults. See "Global QoS Defaults" on page 24-11 for a list of the global defaults. See "Configuring Global QoS Parameters" on page 24-15 for information about configuring global parameters.

2 Configuring QoS Port Parameters. This configuration includes setting up QoS parameters on a per port basis. Typically you will not need to change the port defaults. See "QoS Port Defaults" on page 24-12 for a list of port defaults. See "QoS Ports and Queues" on page 24-24 for information about configuring port parameters.

3 Setting Up Policies. Most QoS configuration involves setting up policies. See "Creating Policies" on page 24-37.

4 Applying the Configuration. All policy rule configuration and some global parameters must be specifically applied through the **qos apply** command before they are active on the switch. See "Applying the Configuration" on page 24-62.

Configuring Global QoS Parameters

This section describes the global QoS configuration, which includes enabling and disabling QoS, applying and activating the configuration, controlling the QoS log display, and configuring QoS port and queue parameters.

Enabling/Disabling QoS

By default QoS is enabled on the switch. If QoS policies are configured and applied, the switch will attempt to classify traffic and apply relevant policy actions.

To disable the QoS, use the **qos** command. For example:

```
-> qos disable
```

QoS is immediately disabled. When QoS is disabled globally, any flows coming into the switch are not classifed (matched to policies).

To re-enable QoS, enter the **qos** command with the **enable** option:

```
-> qos enable
```

QoS is immediately re-enabled. Any policies that are active on the switch will be used to classify traffic coming into the switch.

Note that individual policy rules may be enabled or disabled with the **policy rule** command.

Setting the Global Default Dispositions

By default, bridged, routed, and multicast flows that do not match any policies are accepted on the switch. To change the global default disposition (which determines whether the switch will accept, deny, or drop the flow), use the desired disposition setting (**accept**, **drop**, or **deny**) with any of the following commands: **qos default bridged disposition**, **qos default routed disposition**, or **qos default multicast disposition**.

In the current release, the **drop** and **deny** options produce the same result (flows are silently dropped; no ICMP message is sent).

For example, to deny any routed flows that do not match policies, enter:

```
-> qos default routed disposition deny
```

To activate the setting, enter the **qos apply** command. For more information about the **qos apply** command, see "Applying the Configuration" on page 24-62.

Typically, the disposition is only configured when you are using policies for Access Control Lists (ACLs).

Note that if you set **qos default bridged disposition** to **deny**, you effectively drop all Layer2 traffic that does not match any policy. If you want to create ACLs to allow some Layer 2 traffic through the switch, you must configure two rules for each type of Layer 2 traffic, one for source and one for destination. For more information about ACLs, see Chapter 25, "Configuring ACLs."

Using the QoS Log

The QoS software in the switch creates its own log for QoS-specific events. You may modify the number of lines in the log or change the level of detail given in the log. The PolicyView application, which is used to create QoS policies stored on an LDAP server, may query the switch for log events; or log events can be immediately available to the PolicyView application via a CLI command. Log events may also be forwarded to the console in real time.

What Kind of Information Is Logged

The **debug qos** command controls what kind of information will be displayed in the log. The **qos log level** command determines how specific the log messages will be. See "Log Detail Level" on page 24-17.

By default, only the most basic QoS information is logged. The types of information that may be logged includes rules, Layer 2 and Layer 3 information, etc. For a detailed explanation about the types of information that may be logged, see the *OmniSwitch CLI Reference Guide*. A brief summary of the available keywords is given here:

debug qos keywords		
info	**mem**	**nat**
config	**cam**	**sem**
rule	**mapper**	**pm**
main	**flows**	**ingress**
route	**queue**	**egress**
hre	**slot**	**rsvp**
port	**l2**	**balance**
msg	**l3**	**nimsg**
sl	**classifier**	

To display information about any QoS rules on the switch, enter the **debug qos** command with the **rules** keyword:

```
-> debug qos rules
```

To change the type of debugging, use **no** with the relevant type of information that you want to remove. For example:

```
-> debug qos no rules
```

To turn off debugging (which effectively turns off logging), enter the following:

```
-> no debug qos
```

Enter the **qos apply** command to activate the setting.

Number of Lines in the QoS Log

By default the QoS log displays a maximum of 256 lines. To change the maximum number of lines that may display, use the **qos log lines** command and enter the number of lines. For example:

```
-> qos log lines 30
```

The number of lines in the log is changed. To activate the change, enter the **qos apply** command.

Note. If you change the number of log lines, the QoS log may be completely cleared. To change the log lines without clearing the log, set the log lines in the **boot.cfg** file; the log will be set to the specified number of lines at the next reboot.

Log Detail Level

To change the level of detail in the QoS log, use the **qos log level** command. The log level determines the amount of detail that will be given in the QoS log. The **qos log level** command is associated with the **qos debug** command, which determines what kind of information will be included in the log.

The default log level is 6. The range of values is 1 (lowest level of detail) to 9 (highest level of detail). For example:

```
-> qos log level 7
```

The log level is changed immediately but the setting is not saved in flash. To activate the change, enter the **qos apply** command. For more information about the **qos apply** command, see "Applying the Configuration" on page 24-62.

Note. A high log level value will impact the performance of the switch.

Forwarding Log Events to PolicyView

In addition to managing policies created directly on the switch, the switch manages policies downloaded from an external LDAP server. These policies are created externally through the PolicyView NMS application. PolicyView may query the switch for logged QoS events. Use the **qos forward log** command to make QoS log events available to PolicyView in real time. For example:

```
-> qos forward log
```

To disable log forwarding, enter the following command:

```
-> qos no forward log
```

To activate the change, enter the **qos apply** command. For more information about the **qos apply** command, see "Applying the Configuration" on page 24-62.

If event forwarding is disabled, PolicyView will still be able to query the QoS software for events, but the events will not be sent in real time.

Forwarding Log Events to the Console

QoS log messages may be sent to a console attached directly to the switch. By default, QoS log messages are not sent to the console. To send log events to the console, enter the following command:

```
-> qos log console
```

To disable immediate forwarding of events to the console, enter the following command:

```
-> qos no log console
```

To activate the change, enter the **qos apply** command. For more information about the **qos apply** command, see "Applying the Configuration" on page 24-62.

Displaying the QoS Log

To view the QoS log, use the **show qos log** command. The display is similar to the following:

```
**QOS Log**

Insert rule 0
Rule index at 0
Insert rule 1
Rule index at 1
Insert rule 2
Rule index at 2
Enable rule r1 (1) 1,1
Enable rule r2 (0) 1,1
Enable rule yuba1 (2) 1,1
Verify rule r1(1)
Enable rule r1 (1) 1,1
Really enable r1
Update condition c1 for rule 1 (1)
Verify rule r2(1)
Enable rule r2 (0) 1,1
Really enable r2
Update condition c2 for rule 0 (1)
Verify rule yuba1(1)
Enable rule yuba1 (2) 1,1
Really enable yuba1
Update condition yubamac for rule 2 (1)
QoS Manager started TUE MAR 10 13:46:50 2002

Match rule 2 to 1
Match rule 2 to 2
Match rule 2 to 3
```

The log display may be modified through the **qos log lines**, **qos log level**, and **debug qos** commands. The log display may also be output to the console through the **qos log console** command or sent to the policy software in the switch (which manages policies downloaded from an LDAP server) through the **qos forward log** command.

Clearing the QoS Log

The QoS log can get large if invalid rules are configured on the switch, or if a lot of QoS events have taken place. Clearing the log makes the file easier to manage.

To clear the QoS log, use the **qos clear log** command. For example:

```
-> qos clear log
```

All the current lines in the QoS log are deleted.

Configuring Global Timeouts

Timeouts are used to free up switch resources that maintain flow tables or anticipate reflexive flows. The commands used to configure these timeouts are:

qos flow timeout
qos reflexive timeout
qos nat timeout

Flow Timeout

An entry is made in the flow table whenever a flow is received on the switch. If no packets in the flow are received before the timeout expires, the switch removes the flow entry from the table. Because flow tables take up switch memory, the timeout prevents inactive flow entries from using switch memory. By decreasing the wait time, you can free some memory that the switch is using to keep track of flows; the default value is 300 seconds.

To change the flow timeout, enter the **qos flow timeout** command with the desired number of seconds. For example:

```
-> qos flow timeout 100
```

The timeout will not be active on the switch until you enter the **qos apply** command. For more information about the **qos apply** command, see "Applying the Configuration" on page 24-62.

Reflexive Timeout

When reflexive policy rules are configured and traffic that matches a reflexive rule arrives on the switch, the switch will wait for the reverse flow. These reverse flows may be packets from a TCP session or a filtered IP session. The default timeout is 300 seconds.

To change the timeout, enter the **qos reflexive timeout** command with the desired number of seconds. For example:

```
-> qos reflexive timeout 200
```

The timeout will not be active on the switch until you enter the **qos apply** command. For more information about the **qos apply** command, see "Applying the Configuration" on page 24-62.

For more information about reflexive flows, see Chapter 25, "Configuring ACLs."

NAT Timeout

When address translation rules are configured for many addresses to one address (sometimes referred to as Network Address Translation), the reverse flow will time out after 3600 seconds. Use the NAT timeout to change the amount of time the switch will wait for address translation traffic. Enter the **qos nat timeout** command with the desired number of seconds. For example:

```
-> qos nat timeout 500
```

The timeout will not be active on the switch until you enter the **qos apply** command. For more information about the **qos apply** command, see "Applying the Configuration" on page 24-62.

Fragment Classification

By default, fragments are not classified. The commands available for configuring fragment classification are listed here:

qos classify fragments
qos fragment timeout

When an IP packet reaches a hop with an MTU smaller than the size of the packet, it may be fragmented. If the IP packet contains a TCP or UDP packet, then the TCP/UDP header is copied to the first IP fragment. The remaining fragments contain only user data.

Some policies require that a TCP or UDP packet be classified based on TCP/UDP port number.

The switch has a default policy for fragments, which is to accept all fragments; however, a packet with a fragment offset of 1 will be dropped. IP packets with a fragment offset of 1 are typically used for security attacks.

Enabling/Disabling Fragment Classification

To enable fragment classification, enter the **qos classify fragments** command:

```
-> qos classify fragments
```

The switch will now classify each fragment in the flow.

To disable fragment classification, enter the following command:

```
-> qos no classify fragments
```

The setting is not active until the **qos apply** command is entered. For more information about the **qos apply** command, see "Applying the Configuration" on page 24-62.

Setting the Fragment Timeout

If fragment classification is enabled, the switch waits for all fragments of a packet to arrive. By decreasing the wait time, you can free some memory that the switch is using to keep track of packets; the default value is 10 seconds.

To change the fragment timeout, enter the **qos fragment timeout** command with the desired number of seconds. For example:

```
-> qos fragment timeout 5
```

The timeout will not be active on the switch until you enter the **qos apply** command. (For more information about the **qos apply** command, see "Applying the Configuration" on page 24-62.) The timeout does not take effect if the **qos classify fragments** command has not been entered.

Classifying Bridged Traffic as Layer 3

In some network configurations you may want to force the switch to classify bridged (Layer 2) traffic as routed (Layer 3) traffic. Typically this option is used for QoS filtering. See Chapter 25, “Configuring ACLs,” for more information about filtering.

If this option is enabled:

- Switch performance may be slower.
- The switch may bridge and route traffic to the same destination.

Note. When **qos classifyl3 bridged** is enabled, bridged IP packets are prioritized based on ToS, not 802.1p.

If **qos classifyl3 bridged** is enabled, the switch will classify bridged IP packets with Layer 3 (routing) conditions; however, Layer 2 source and destination rules take precedence. If the default routed disposition is set to **deny** or **drop** when **qos classifyl3 bridged** is enabled, all bridged IP packets will be dropped.

To configure the switch to classify Layer 2 traffic as Layer 3, use the **qos classifyl3 bridged** command.

```
-> qos classifyl3 bridged
```

To disable classifying Layer 2 traffic as Layer 3, enter the **no** form of the command:

```
-> qos no classifyl3 bridged
```

The setting is not active until you enter the **qos apply** command. For more information about the **qos apply** command, see “Applying the Configuration” on page 24-62.

Setting the Statistics Interval

To change how often the switch polls the network interfaces for QoS statistics, use the **qos stats interval** command with the desired interval time in seconds. The default is 60 seconds. For example:

```
-> qos stats interval 30
```

Statistics are displayed through the **show qos statistics** command. For more information about this command, see the *OmniSwitch CLI Reference Guide*.

Returning the Global Configuration to Defaults

To return the global QoS configuration to its default settings, use the **qos reset** command. The defaults will then be active on the switch. For a list of global defaults, see “QoS Defaults” on page 24-11.

Note. The **qos reset** command only affects the global configuration. It does not affect any policy configuration.

Verifying Global Settings

To display information about the global configuration, use the following **show** commands:

show qos config	Displays global information about the QoS configuration.
show qos statistics	Displays statistics about QoS events.

For more information about the syntax and displays of these commands, see the *OmniSwitch CLI Reference Guide*.

QoS Ports and Queues

Queue parameters may be modified on a port basis; for high-density gigabit modules, they may be modified per *slice* as well. (A slice is a logical section of hardware and corresponds to particular ports on a network interface module.)

Default Queues

Four default queues are created for each port on the switch at start up. These queues service four internal priorities: low (0), medium (1), high (2), and urgent or highest (3). A packet that does not match a QoS policy is placed into one of these queues based on the packet's internal priority, which the switch derives from the packet's 802.1p value (bridged traffic) or ToS/DSCP value (routed traffic).

The packet's 802.1p or ToS/DSCP value is affected by the following:

- What the default 802.1p or ToS/DSCP setting is for the port. See "Configuring the 802.1p or ToS/DSCP Port Default" on page 24-25.
- Whether or not the port is trusted (able to recognize 802.1p/ToS/DSCP values in incoming traffic). For more information about trusted ports, see "Trusted and Untrusted Ports" on page 24-26.

Queues Created for Flows That Match Policies

The switch creates additional queues based on policy rules that match incoming flows. When a flow coming into the switch matches a policy, it is queued based on:

- Parameters given in the policy action (specified by the **policy action** command) with any of the following keywords: **priority**, **shared**, **minimum bandwidth**, **maximum bandwidth**, or **maximum buffers**.
- Port settings configured through the **qos port** command.

The priority of the queue depends on the policy rule. If no priority is given in the rule, the switch will assume priority 0 (lowest priority).

Note. Flows coming into the switch that match policies are placed in QoS queues; flows coming into the switch that do not match policies are placed in default queues. The exception is a flow matching a policy that specifies a disposition action of **accept** and no other action parameters are configured; that flow will be placed in a default queue.

Shared Queues

A separate queue is created for each flow. Flows may share queues if they match the same policy and the policy action is configured for sharing. You may want to configure queue sharing to preserve resources, or you might want to place flows with the same requirements into the same queue. For example, you might want to force flows that require 5 Mbps to share a 10 Mbps queue rather than have a 5 Mbps queue created for each flow.

In order to share queues, ingress traffic must meet the following requirements:

- Traffic must be sent over the same slice on the same network interface module (or slot).
- Traffic must egress out the same physical port.

A slice is a logical section of hardware that corresponds to particular ports on a network interface module. On the OmniSwitch 7700/7800, each network interface module contains a single slice (slice 0); all ports on a network interface module are on the same slice. On the OmniSwitch 8800, however, each network interface module may contain up to four slices (slices 0 to 3); ports on the same interface module may be on different slices. Only ingress traffic sent over the same slice may share queues. Traffic arriving on the same interface but on different slices cannot share queues.

If flows coming in on the OmniSwitch 8800 match a policy configured for sharing queues, and the flows all arrive on slice 0 of slot 2, a single queue is created and the queue is shared. If the flows instead come in on slice 0 and slice 1 of slot 2, the switch creates separate queues, and queue sharing does not take place.

In the following example, two policy rules are created that classify traffic with different source IP addresses. The policy rules use the same action for maximum bandwidth. First, two conditions are created (**net1** and **net2**) for the source IP addresses. Then the action for maximum bandwidth is created (**act40**); initially the action is not configured to share queues. Then each condition is associated with the **act40** action in two policy rules (**net1max** and **net2max**). The **qos apply** command is entered to activate the configuration.

```
-> policy condition net1 source ip 198.60.82.0 mask 255.255.255.0
-> policy condition net2 source ip 143.209.92.0 mask 255.255.255.0
-> policy action act4 maximum bandwidth 40m
-> policy rule net1max condition net1 action act4
-> policy rule net2max condition net2 action act4
-> qos apply
```

In this scenario, when traffic flows match **net1max** and **net2max**, the switch will create two separate queues, each with a maximum bandwidth of 40 Mbps. To configure **act4** as shared, enter the **policy action shared** command:

```
-> policy action act4 shared
-> qos apply
```

When flows come into the switch that match **net1max** and **net2max**, the switch will create a single queue with a maximum bandwidth of 40 Mbps. Note that the flows must arrive on the same slice and be destined for the same egress port in order to share the queue.

For more information about configuring policy actions, see "Creating Policy Actions" on page 24-40. For more information about configuring policy rules, see "Creating Policy Rules" on page 24-41.

Configuring the 802.1p or ToS/DSCP Port Default

Ports handle 802.1p bits or ToS/DSCP values differently depending on whether the ports are trusted or untrusted and what the default 802.1p or ToS/DSCP setting is for the port. For more information about trusted and untrusted ports, see "Trusted and Untrusted Ports" on page 24-26.

By default, the port defaults for 802.1p and ToS/DSCP are 0. To change the default 802.1p or ToS/DSCP setting for a port, use the **qos port default 802.1p** or **qos port default dscp** command. For example:

```
-> qos port 3/1 default 802.1p 7
```

In this example, any untagged traffic (traffic without any 802.1p setting) arriving on port 3/1 will be stamped with an 802.1p value of 7 (highest priority). If the port is configured to be untrusted, any tagged traffic will be stamped with an 802.1p value of 7. If the port is configured to be trusted, any tagged traffic will preserve the 802.1p value in the flow.

If the flow arriving on port 3/1 matches a policy that specifies an 802.1p action, the 802.1p value specified in the action will be used if the port is trusted; if the port is untrusted, the default port value will be used.

To activate the change, enter the **qos apply** command. For more information about the **qos apply** command, see "Applying the Configuration" on page 24-62. See the next section for more details about trusted and untrusted ports.

Trusted and Untrusted Ports

By default switch ports are *not trusted*; that is, they do not recognize 802.1p or ToS/DSCP settings in packets of incoming traffic. If a port is not trusted, the switch sets any 802.1p or ToS/DSCP to zero in incoming packets. The switch may be set globally so that all ports will recognize 802.1p bits. Individual ports may be configured to override the global setting.

Note. Mobile ports and ports that are configured for 802.1Q are always trusted ports regardless of the QoS settings. These ports cannot be configured as untrusted.

The difference between trusted and untrusted ports is described here:

Trusted Ports

- The 802.1p or ToS/DSCP value is preserved.
- The VLAN specified for the flow is also preserved.
- If the incoming 802.1p or ToS/DSCP flow does not match a policy, the switch places the flow into a default queue and prioritizes the flow based on the 802.1p or ToS/DSCP value in the flow.
- If the incoming 802.1p or ToS/DSCP flow matches a policy, the switch queues the flow based on the policy action.

Untrusted Ports

- The switch zeroes out the 802.1p bits or ToS/DSCP value.
- The VLAN is reset to the default VLAN on the port.
- If the incoming 802.1p or ToS/DSCP flow does not match a policy, the switch places the flow into a default queue based on the port default for 802.1p (for bridged traffic) or the port default for ToS/DSCP (for routed traffic).
- If the incoming 802.1p or ToS/DSCP flow matches a policy, the switch places the flow into a QoS queue based on the port default for 802.1p (for bridged traffic) or the port default for ToS/DSCP (for routed traffic).

The OmniSwitch does not allow 802.1p or ToS/DSCP settings in a policy action to be stamped on a classified flow ingressing on an untrusted port. Note that the **qos port default 802.1p** or the **qos port default dscp** command may be used to configure the port so that it will set 802.1p or ToS/DSCP on flows going through a particular port. See "Configuring the 802.1p or ToS/DSCP Port Default" on page 24-25 for more information about using these commands.

Configuring Trusted Ports

By default, all ports (except 802.1Q-tagged ports and mobile ports) are untrusted. The trust setting may be configured globally on the switch, or on a per-port basis.

To configure the global setting on the switch, use the **qos trust ports** command. For example:

```
-> qos trust ports
```

To configure individual ports to recognize 802.1p or ToS, use the **qos port trusted** command with the desired slot/port number. For example:

```
-> qos port 3/2 trusted
```

The global setting is active immediately; however, the port setting requires qos apply to activate the change. For more information about the **qos apply** command, see "Applying the Configuration" on page 24-62.

Using Trusted Ports With Policies

Whether or not the port is trusted is important if you want to classify traffic with 802.1p bits. If the policy condition specifies 802.1p, the switch must be able to recognize 802.1p values. 802.1p bits may be set or mapped to a single value using the **policy action 802.1p** command. In this example, the **qos port** command specifies that port 2 on slot 3 will be able to recognize 802.1p bits. A policy condition (**Traffic**) is then created to classify traffic containing 802.1p bits set to 4 and destined for port 2 on slot 3. The policy action (**SetBits**) specifies that the bits will be reset to 7 when the traffic egresses the switch. A policy rule called **Rule2** puts the condition and the action together.

```
-> qos port 3/2 trusted
-> policy condition Traffic destination port 3/2 802.1p 4
-> policy action SetBits 802.1p 7
-> policy rule Rule2 condition Traffic action SetBits
```

To activate the configuration, enter the **qos apply** command. For more information about the **qos apply** command, see "Applying the Configuration" on page 24-62.

For actions that set 802.1p bits, note that a limited set of policy conditions are supported. For information about which conditions may be used with an 802.1p action, see "Condition/Action Combinations" on page 24-8.

Note. 802.1p mapping may also be set for Layer 3 traffic, which typically has the 802.1p bits set to zero.

QoS and High-Density Gigabit Modules

When traffic ingresses the switch through ports on high-density gigabit modules, the traffic is classified differently than on other switch interfaces. On these modules, packets coming into the switch are classified on the module first, before any QoS settings take effect.

IP traffic on high-density ports may be classified by ToS or DSCP, or 802.1p.

In addition, high-density gigabit modules have several other QoS settings for determining the internal protocol priority and controlling the buffering, queuing, servicing, and discarding of packets.

For more information about the commands described in this section, see the *OmniSwitch CLI Reference Guide*.

Terminology Note

There are several terms that are used in describing the configuration options for QoS on high-density gigabit modules. These are as follows:

- **Slice**—A *slice* is a logical section of hardware that corresponds to particular ports on a network interface module. On the OmniSwitch 7700/7800, each network interface module may have one slice (0). On the OmniSwitch 8800, each network interface module may have up to 4 slices (0 to 3).

  To view information about which ports are included in a slot/slice on the OmniSwitch 8800, use the **show policy port group** command to view the built-in port groups.

- **Chunk**—A *chunk* is 64 bytes of the 256-Kbyte buffer on the high-density interface (4096 possible chunks). Every received packet on the high-density interface may be stored in a 64-byte chunk of the buffer.

Default Classification

If your switch has high-density gigabit modules installed, you may want to change the default classification of ports on those modules for *IP traffic only* using the **qos port default classification** command. The setting does not affect Layer 2 traffic on these ports, which is always classified with 802.1p.

Traffic ingressing on untrusted ports is stamped with 802.1p/ToS/DSCP values as specified by the **qos port default 802.1p** and **qos port default dscp** commands. (By default these values are zero.) The default setting specified by **qos port default classification** is then used to prioritize the flow.

To change the default classification of a port on a high-density gigabit module, use the **qos port default classification** command with the relevant port number and the desired classification (**802.1p**, **tos**, or **dscp**). For example:

```
-> qos port 3/1 default classification tos
```

IP traffic received on the high-density gigabit interface through port 3/1 will now be classified by ToS rather than 802.1p. Any Layer 2 traffic received on the port will still be classified with 802.1p.

Controlling Internal Priority

The switch uses four internal priority queues for each interface. These internal priorities are specified as 0 (lowest priority) to 3 (highest priority). A packet coming into the switch on a high-density port is mapped to an internal priority based on the ingress priority value and the protocol in the packet.

Note. The **qos port default classification** command determines how packets are classified for any protocols not specified by the **qos port protocol priority** command.

The Protocol CAM

The high-density module has a protocol CAM it uses to match the incoming packet protocol. Each entry in the CAM includes a protocol ID and protocol type. It may also include destination Service Access Point (SAP) and source SAP codes, whether or not 802.3 is enabled or disabled, and precedence information for the entry. The protocol CAM has built-in entries; manual entries may be created through the **qos slice** command.

When the protocol CAM matches the protocol in the incoming packet, it then indexes the packet with an internal priority that is used for queuing the packet. If a matching protocol cannot be found (unknown protocol), the default classification and priority is used for the packet. The default priority is determined by the **qos port default 802.1p** or **qos port default dscp** commands.

Up to four user-configured protocol entries may be configured per slot/slice. Two of these may have a precedence of **priority**, which means the entries take precedence over built-in entries. The other two may have a precedence of **fallback**, which means the built-in CAM entries take precedence.

Note. To view the built-in CAM entries, as well as any manually configured entries, use the **show qos slice pcams** command.

To create a user-configured protocol CAM entry on a slot/slice, use the **qos slice** command with the relevant slot/slice number, the protocol ID, and the Ethertype. Optionally you may specify the destination SAP and source SAP values, the precedence, and whether 802.3 is enabled or disabled. For example:

```
-> qos slice 3/0 protocol user1 ethertype 0800 dsapssap aa/aa priority
```

In this example, the value **user1** indicates that this is the first user-configured protocol CAM entry on slice 0 of slot 3. The entry is configured with an Ethertype of **0800** (IP), a destination SAP value and source SAP value of **aa** (SNAP). The entry is configured with the **priority** keyword, which means it will take precedence over built-in entries. When a packet comes into the switch that matches both a built-in entry and this user-configured entry, the switch will use this user-configured entry to index the packet's internal priority.

Protocol Priority for 802.1p/ToS Traffic

By default, the internal priority for 802.1p or ToS packets is mapped by the switch as follows:

ToS values (ingress priority)	internal priority
0–1	0
2–3	1
4–5	2
6–7	3

The internal priority may be controlled on a given slot/port for a particular protocol type through the **qos port protocol priority** command. (The current internal priority configuration may be displayed through the **show qos port pdis** command.)

Note. The port must be trusted or the protocol priority will not take effect. See "Trusted and Untrusted Ports" on page 24-26 for more information about trusted ports.

The built-in protocol priority mapping for IPv4, LACP, and BPDU traffic cannot be modified. However, user-specified protocols may be configured to override the built-in mapping.

For example:

```
-> qos port 1/3 protocol user1 priority 0 0 1 1 1 2 3 3 classification tos
```

In this example, the protocol for user1 is defined as IP traffic through the **qos slice** command (described in "The Protocol CAM" on page 24-29). The **qos port protocol priority** command given in the example then specifies that IP traffic coming in on port 3 of slot 1 will be classified by ToS priority and mapped to internal priorities as follows:

ToS values (ingress priority)	internal priority
0–1	0
2–4	1
5	2
6–7	3

Protocol Priority for DSCP Traffic

The switch uses a different internal table for mapping DSCP values to internal priority. The default table is as follows:

DSCP index	value (internal priority)
0–15	0
16–31	1
32–47	2
48–63	3

The table mapping for DSCP is controlled per slot/slice (rather than slot/port) with the **qos slice dscp** command. For example:

```
-> qos slice 2/0 dscp 30 2
```

In this example, any DSCP traffic coming into the switch on slice 0 of slot 2 with a DSCP value of 30 would get mapped to internal priority 2 (rather than the default of internal priority 1).

Controlling Queuing and Discarding at the Port Level

High-density ports have configurable First In First Out (FIFO) thresholds, which determine how incoming packets are enqueued and discarded on the port. The thresholds each have an upper and lower value for each internal priority queue on the port.

The thresholds specify a count of the number of "chunks" of buffer for packets with a particular internal priority. (For the definition of "chunk," see "Terminology Note" on page 24-28.)

When the upper threshold is reached, packets with that internal priority will be discarded. When the lower threshold is reached, packets with that internal internal priority will be enqueued rather than discarded.

To configure enqueuing thresholds, use the **qos port enqueuing thresholds** command with the desired upper and lower thresholds for each priority queue. The thresholds are ordered as follows: *up0-low0 up1-low1 up2-low2 up3-low3*. For example:

```
-> qos port 1/2 enqueuing thresholds 30-10 35-10 55-10 500-250
```

In this example, the enqueuing thresholds on port 2 of slot 1 for priority queue 0 are 30 (upper) and 10 (lower). Packets will be enqueued as follows:

- When packets with internal priority 0 arrive on the port, they will be enqueued until the upper threshold is reached (30 chunks of buffer are full). When 30 chunks have accumulated, any additional packets with priority 0 are discarded. When the buffer is depleted to the lower threshold (10 chunks), packets with priority 0 will be enqueued until the upper threshold is reached again.

- When packets with internal priority 1 arrive on the port, they will be enqueued until the upper threshold is reached (35 chunks of buffer are full). When 35 chunks have accumulated, any additional packets with priority 1 are discarded. When the buffer is depleted to the lower threshold (10 chunks), packets with priority 1 will be enqueued until the upper threshold is reached again.

- When packets with internal priority 2 arrive on the port, they will be enqueued until the upper threshold is reached (55 chunks of buffer are full). When 55 chunks have accumulated, any additional packets with priority 2 are discarded. When the buffer is depleted to the lower threshold (10 chunks), packets with priority 2 will be enqueued until the upper threshold is reached again.

- When packets with internal priority 3 arrive on the port, they will be enqueued until the upper threshold is reached (500 chunks of buffer are full). When 500 chunks have accumulated, any additional packets with priority 3 are discarded. When the buffer is depleted to the lower threshold (250 chunks), packets with priority 3 will be enqueued until the upper threshold is reached again.

The **qos port enqueuing thresholds** command controls discarding on a particular *port*. For controlling discarding on a *slice* (which is essentially a group of ports), see "Controlling Discarding on the Slice (WRED)" on page 24-34.

Important. When the 256 Kbyte buffer is completely full, all packets coming in will be discarded until at least one 64-byte chunk is free.

How Queues are Serviced on High-Density Modules

Once packets are enqueued on a high-density *slice* (basically, a group of ports), they are dequeued based on a standard servicing algorithm (or *mode*) or user-configured servicing thresholds. The servicing modes are a way of automatically setting servicing thresholds.

Note. For the definition of “slice,” see “Terminology Note” on page 24-28.

Servicing Modes

The default servicing mode is Priority Weighted Round Robin; other possible modes are strict priority and Weighted Round Robin.

Each servicing algorithm uses priority as a basis for sending out packets:

- **Strict Priority**—Packets with the highest priority are always sent first.
- **Weighted Round Robin**—Each priority queue gets serviced on every “round,” with higher priorities sending more packets than lower priority queues.
- **Priority Weighted Round Robin**—Each0 priority queue gets serviced on every “round,” with higher priorities sending more packets than lower priorities until a given threshold is reached. When a threshold is reached, the sequence of sending packets is changed so that only packets of the priority that crossed the threshold are sent out. When the buffer is below the threshold again, packets will be sent out of every priority queue again.

Each threshold specifies a maximum number of chunks of buffer for packets of a particular internal priority. (For the definition of “chunk,” see “Terminology Note” on page 24-28.)

The thresholds that these modes automatically set up are given here:

Servicing Mode	**p0**	**p1**	**p2**	**p3**
Priority Weighted Round Robin (default)	45	90	180	360
Strict Priority	0	0	0	0
Weighted Round Robin	4095	4095	4095	4095

Note. The p0 threshold is not configurable; it is always set automatically by the switch based on the servicing algorithm.

To change the servicing mode, use the **qos slice servicing mode** command with the desired keyword for the mode (**strict-priority**, **priority-wrr**, or **wrr**). For example:

```
-> qos slice 2/0 servicing mode strict-priority
```

In this example, the servicing mode of slice 0 on slot 2 is set to strict priority, which means that highest priority packets will always be sent out first.

If the mode is set to Weighted Round Robin (**wrr**):

```
-> qos slice 2/0 servicing mode wrr
```

Then the thresholds are set to the top of the data buffer, and the priority queues are serviced based on the Weighted Round Robin algorithm in the switch, which send packets out by alternating packets of different priority, giving weight to higher priority packets.

User-Configured Servicing Thresholds

You can configure the servicing thresholds for each priority queue using the **qos slice servicing mode** command with the **priority-wrr** keyword. The thresholds are ordered as follows: *p1 p2 p3*.

```
-> qos slice 2/0 servicing mode priority-wrr 80 200 300
```

In this example, on slice 0 of slot 2 the servicing threshold for the priority 1 queue is 80. When 80 chunks of buffer have accumulated with packets of priority 1, packets with priority 1 will be sent out. Likewise, for the other priority queue thresholds, once the threshold is reached (by accumulating packets of that priority into buffer chunks), packets with that particular priority are sent out. (The priority 0 threshold is always 45; it is not configurable.)

Controlling Discarding on the Slice (WRED)

The buffer accumulation thresholds on the microchip of the high-density module may also be manually modified to use the Weighted Random Early Detection (WRED) algorithm for discarding packets. By default, WRED is disabled on each high-densisty gigabit slice, and all thresholds are set to the maximum amount of chunks of buffer that may be used (4095).

Note. For the definition of "chunk," see "Terminology Note" on page 24-28.

When any of the WRED thresholds are modified to be anything other than zero, this effectively enables WRED on the slice. The WRED algorithm uses the the current chunk count and a weight value to determine the average chunk count; the average chunk count is used to determine when the WRED thresholds are reached. The weight value (a negative power of 2) is configurable; the default is 4.

Each priority queue has a WRED threshold range. When the lower threshold is reached, some frames with that priority are discarded. When the upper threshold is reached, all frames with that priority are discarded. As the average chunk count gets closer to the upper threshold, more frames are discarded. The decision to discard is made randomly by the switch.

To set WRED thresholds, use the **qos slice wred thresholds** command; the order of the thresholds is as follows: *up0-low0 up1-low1 up2-low2 up3-low3*. For example:

```
-> qos slice 3/0 wred thresholds 30-10 35-10 40-10 575-260 weight 5
```

In this example, a weight value of 2^{-5} will be used to compute the average chunk count. Discarding will happen as follows:

- For priority queue 0, when enough packets with priority 0 have accumulated to meet the lower threshold (10), some packets with priority 0 will be discarded. When the upper threshold (30) is reached, all packets with priority 0 will be discarded.
- For priority queue 1, when enough packets with priority 1 have accumulated to meet the lower threshold (10), some packets with priority 1will be discarded. When the upper threshold (35) is reached, all packets with priority 1 will be discarded.
- For priority queue 2, when enough packets with priority 2 have accumulated to meet the lower threshold (10), some packets with priority 2 will be discarded. When the upper threshold (40) is reached, all packets will priority 2 will be discarded.

- For priority queue 3, when enough packets with priority 3 have accumulated to meet the lower threshold (260), some packets with priority 3 will be discarded. When the upper threshold (575) is reached, all packets with priority 3 will be discarded.

In all cases, as the average chunk count gets closer to the upper threshold, more packets are discarded randomly.

The **qos slice wred thresholds** command controls discarding on a particular *slice*. For controlling discarding on an individual *port*, see "Controlling Queuing and Discarding at the Port Level" on page 24-32.

Important. When the 256-Kbyte buffer is completely full, all packets coming in will be discarded until at least one 64 byte chunk is free.

Verifying the QoS Port and Queue Configuration

To display information about QoS ports and queues, use the following commands:

show qos port Displays information about all QoS ports or a particular port. Use the **statistics** option of this command to get information about actual bytes and packets queued, dequeued, and discarded on high-density ports.

show qos port high-density-module Displays information about the enqueuing thresholds on high-density QoS ports.

show qos port pdis Displays priority descriptors for slot/ports on high-density modules.

show qos queue Displays information for all QoS queues or only those queues associated with a parti29
cular slot/port.

show qos slice Displays information about buffers on QoS slices of QoS slots.

show qos slice high-density-module Displays information about servicing thresholds and WRED thresholds for slots/slices on high-density gigabit modules.

show qos slice pcams Displays the current CAM entries for high-density modules on the switch.

See the *OmniSwitch CLI Reference Guide* for more information about the syntax and displays for these commands.

Creating Policies

This section describes how to create policies in general. For information about configuring specific types of policies, see "Policy Applications" on page 24-65.

Basic commands for creating policies are as follows:

policy condition
policy action
policy rule

This section describes generally how to use these commands. See "Policy Applications" on page 24-65 for information about configuring specific types of policies. For additional details about command syntax, see the *OmniSwitch CLI Reference Guide*.

Note. A policy rule may include a policy condition or a policy action that was created through PolicyView rather than the CLI. But a policy rule, policy action, or policy condition may only be modified through the source that created it. For example, if an action was created in PolicyView, it may be included in a policy rule configured through the CLI, but it cannot be modified through the CLI.

Policies are not used to classify traffic until the **qos apply** command is entered. See "Applying the Configuration" on page 24-62.

To view information about how the switch will classify particular condition parameters, use the **show policy classify** command. This is useful to test conditions before actually activating the policies on the switch. See "Testing Conditions" on page 24-47.

Quick Steps for Creating Policies

Follow the steps below for a quick tutorial on creating policies. More information about how to configure each command is given in later sections of this chapter.

1 Create a policy condition with the **policy condition** command. For example:

```
-> policy condition cond3 source ip 10.10.2.3
```

Note. (*Optional*) Test the rule with the **show policy classify** command using information from the policy condition. For example:

```
-> show policy classify l3 source ip 10.10.2.3
```

This command displays information about whether or not the indicated parameter may be used to classify traffic based on policies that are configured on the switch.

2 Create a policy action with the **policy action** command. For example:

```
-> policy action action2 priority 7
```

3 Create a policy rule with the **policy rule** command. For example:

```
-> policy rule my_rule condition cond3 action action2
```

4 Use the **qos apply** command to apply the policy to the configuration. For example:

```
-> qos apply
```

Note. (*Optional*) To verify that the rule has been configured, use the **show policy rule** command. The display is similar to the following:

```
-> show policy rule
                   Policy          From   Prec   Enabled Inactive Reflexive
my_rule                            cli    0      Yes      Yes     No
Cnd/Act:           cond5 -> action2

+my_rule5                          cli    0      Yes     No      No
Cnd/Act:           cond2 -> pri2

mac1                               cli    0      Yes     No      No
Cnd/Act:           dmac1 -> pri2
```

This command displays information about whether or not the indicated parameter may be used to classify traffic based on policies that are configured on the switch. For more information about this display, see "Verifying Policy Configuration" on page 24-45.

An example of how the example configuration commands might display when entered sequentially on the command line is given here:

```
-> policy condition cond3 source ip 10.10.2.3
-> policy action action2 priority 7
-> policy rule my_rule condition cond3 action action2
-> qos apply
```

ASCII-File-Only Syntax

When the **policy rule**, **policy condition**, and **policy action** commands as well as any of the condition group commands are configured and saved in an ASCII file (typically through the **snapshot** command), the commands included in the file will include syntax indicating the command's origin. The origin specifies where the rule, condition, condition group, or action was created, either an LDAP server or the CLI (**from ldap** or **from cli**). For built-in QoS objects, the syntax displays as **from blt**. For example:

```
-> policy action A2 from ldap disposition accept
```

The **from** option is configurable (for LDAP or CLI only) on the command line; however, it is not recommended that a QoS object's origin be modified. The **blt** keyword indicates built-in; this keyword cannot be used on the command line. For information about built-in policies and QoS groups, see "How Policies Are Used" on page 24-4.

Creating Policy Conditions

This section describes how to create policy conditions in general. Creating policy conditions for particular types of network situations is described later in this chapter.

Note. Policy condition configuration is not active until the **qos apply** command is entered. See “Applying the Configuration” on page 24-62.

To create or modify a policy condition, use the **policy condition** command with the keyword for the type of traffic you want to classify, for example, an IP address or group of IP addresses. In this example, a condition (**c3**) is created for classifying traffic from source IP address 10.10.2.1:

```
-> policy condition c3 source ip 10.10.2.1
```

There are many options for configuring a condition, depending on how you want the switch to classify traffic for this policy. An overview of the options is given here. Later sections of this chapter describe how to use the options in particular network situations.

Note. The group options in this command refer to groups of addresses, services, or ports that you configure separately through policy group commands. Rather than create a separate condition for each address, service, or port, use groups and attach the group to a single condition. See “Using Condition Groups in Policies” on page 24-49 for more information about setting up groups.

More than one condition parameter may be specified. Some condition parameters, like ToS and DSCP, are mutually exclusive. For supported combinations of condition parameters, see “Condition Combinations” on page 24-6.

policy condition keywords
source ip
destination ip
source network group
destination network group
source ip port
destination ip port
source tcp port
destination tcp port
source udp port
destination udp port
service
service group
ip protocol
tos
dscp
source mac
destination mac
source mac group
destination mac group
source vlan
destination vlan
source port
source port group
destination port
destination port group
source interface type
destination interface type

The condition will not be active on the switch until you enter the **qos apply** command.

Removing Condition Parameters

To remove a classification parameter from the condition, use **no** with the relevant keyword. For example:

```
-> policy condition c3 no source ip
```

The specified parameter (in this case, a source IP address) will be removed from the condition (**c3**) at the next **qos apply**.

Note. You cannot remove all parameters from a policy condition. A condition must be configured with at least one parameter.

Deleting Policy Conditions

To remove a policy condition, use the **no** form of the command. For example:

```
-> no policy condition c3
```

The condition (**c3**) cannot be deleted if it is currently being used by a policy rule. If a rule is using the condition, the switch will display an error message. For example:

```
ERROR: c3 is being used by rule 'my_rule'
```

In this case, the condition will not be deleted. The condition (**c3**) must first be removed from the policy rule (**my_rule**). See "Creating Policy Rules" on page 24-41 for more information about setting up rules.

If **c3** is not used by a policy rule, it will be deleted after the next **qos apply**.

Creating Policy Actions

This section describes how to configure policy actions in general. Creating policy actions for particular types of network situations is described later in this chapter.

To create or modify a policy action, use the **policy action** command with the desired action parameter. A policy action should specify the way traffic should be treated. For example, it might specify a priority for the flow, a source address to rewrite in the IP header, or it may specify that the flow may simply be denied. For example:

```
-> policy action Block disposition deny
```

In this example, the action (**Block**) has a disposition of **deny** (disposition determines whether a flow is allowed or denied on the switch). This action may be used in a policy rule to deny a particular type of traffic specified by a policy condition.

Note. Policy action configuration is not active until the **qos apply** command is entered. See "Applying the Configuration" on page 24-62.

More than one action parameter may be specified. Some parameters, like ToS and DSCP, are mutually exclusive. In addition, some action parameters are only supported with particular condition parameters. For information about supported combinations of condition and action parameters, see "Condition/Action Combinations" on page 24-8. See the *OmniSwitch CLI Reference Guide* for details about command syntax.

policy action keywords		
disposition	**tos**	**source rewrite ip**
shared	**dscp**	**source rewrite ip network group**
priority	**map**	**destination rewrite ip**
minimum bandwidth		**destination rewrite ip network group**
maximum bandwidth		
maximum depth		
maximum buffers		

Note. If you combine **priority** with **802.1p**, **dscp**, **tos**, or **map**, in an action, the priority value is used to prioritize the flow.

Removing Action Parameters

To remove an action parameter or return the parameter to its default, use **no** with the relevant keyword.

```
-> policy action a6 no priority
```

This example removes the configured priority value from action **a6**. If any policy rule is using action **a6**, the default action will be to allow the flow classified by the policy condition.

The specified parameter (in this case, priority) will be removed from the action at the next **qos apply**.

Deleting a Policy Action

To remove a policy action, use the **no** form of the command.

```
-> no policy action a6
```

The action cannot be deleted if it is currently being used by a policy rule. If a rule is using the action, the switch will display an error message. For example:

```
ERROR: a6 is being used by rule 'my_rule'
```

In this case, the action will not be deleted. The action (**a6**) must first be removed from the policy rule (**my_rule**). See "Creating Policy Rules" on page 24-41 for more information about setting up rules.

If **a6** is not used by a policy rule, it will be deleted after the next **qos apply**.

Creating Policy Rules

This section describes in general how to create or delete policy rules and rule parameters. See later sections of this chapter for more information about creating particular types of policy rules.

To create a policy rule, use the **policy rule** command and specify the name of the rule, the desired condition, and the desired action.

In this example, condition **c3** is created for traffic coming from IP address 10.10.8.9, and action **a6** is created to allow queue sharing. Policy rule **rule5** combines the condition and the action, so that traffic arriving on the switch from 10.10.8.9 will be placed into a QoS queue that may be shared by other flows.

```
-> policy condition c3 source ip 10.10.8.9
-> policy action a6 shared
-> policy rule rule5 condition c3 action a6
```

The rule (**rule5**) will only take effect after the **qos apply** command is entered. For more information about the **qos apply** command, see "Applying the Configuration" on page 24-62.

The **policy rule** command may specify the following keywords:

policy rule keywords	
precedence	**save**
reflexive	**log**

In addition, a policy rule may be administratively disabled or re-enabled using the **policy rule** command. By default rules are enabled. For a list of rule defaults, see "Policy Rule Defaults" on page 24-12.

Information about using the **policy rule** command options is given in the next sections.

Disabling Rules

By default, rules are enabled. Rules may be disabled or re-enabled through the **policy rule** command using the **disable** and **enable** options. For example:

```
-> policy rule rule5 disable
```

This command prevents **rule5** from being used to classify traffic.

Note that if **qos disable** is entered, the rule will not be used to classify traffic even if the rule is enabled. For more information about enabling/disabling QoS globally, see "Enabling/Disabling QoS" on page 24-15.

Rule Precedence

The switch attempts to classify flows coming into the switch according to policy precedence. The rule with the highest precedence will be applied to the flow.

How Precedence is Determined

Precedence is determined by the following, in order:

- **The type of QoS rule** (Layer 2 source, Layer 2 destination, or Layer 3)—When a flow comes into the switch, the Layer 2 source rules are examined first for a match. If no match is found, the Layer 2 destination rules are examined. If no match is found, the Layer 3 rules are examined.
- **Precedence value**—Each policy has a precedence value. The value may be user-configured through the **policy rule** command in the range from 0 (lowest) to 65535 (highest). (The range 30000 to 65535 is typically reserved for PolicyView.) By default, a policy rule has a precedence of 0. See the next section for information about configuring precedence for a particular rule.
- **Configured rule order**—If a flow matches more than one rule in a particular precedence list (for example, the Layer 2 source list), and both rules have the same precedence value, the rule that was *configured first* in the list will take precedence.

If a flow matches more than one rule in a particular precedence list (for example, the Layer 2 source list), the precedence determines which rule the switch will apply to the flow.

Note. If you configure Layer 2 traffic to be classified as Layer 3 (through the **qos classifyl3 bridged** command), and you have Layer 2 destination rules configured on the switch, make sure the Layer 2 destination rules don't match the traffic that you want to classify as Layer 3. Otherwise, the flows will not be classified as Layer 3 since Layer 2 destination rules take precedence.

Precedence is particularly important for Access Control Lists (ACLs). For more details about precedence and examples about using precedence, see Chapter 25, "Configuring ACLs."

Specifying Precedence for a Particular Rule

To specify a precedence value for a particular rule, use the **policy rule** command with the precedence keyword. For example:

```
-> policy rule r1 precedence 200 condition c1 action a1
```

Creating Reflexive Rules

Reflexive rules enable traffic that would normally be denied on the switch—because another policy denies that traffic—to be allowed through the switch. Typically reflexive policies are used for Access Control Lists (ACLs). For more information about reflexive policies*,* see Chapter 25, "Configuring ACLs."

```
-> policy rule rule5 reflexive
```

To modify a reflexive rule so that it is not reflexive, use **no** with the **reflexive** keyword. For example:

```
-> policy rule rule5 no reflexive
```

Saving Rules

The **save** option marks the policy rule so that the rule will be captured in an ASCII text file (using the **configuration snapshot** command) and saved to the working directory (using the **write memory** command or **copy running-config working** command). By default, rules are saved.

If the **save** option is removed from a rule, the **qos apply** command may activate the rule for the current session, but the rule will not be saved over a reboot. Typically, the **no save** option is used for temporary policies that you do not want saved in the switch configuration file.

To remove the **save** option from a policy rule, use **no** with the **save** keyword. For example:

```
-> policy rule rule5 no save
```

To reconfigure the rule as saved, use the **policy rule** command with the **save** option. For example:

```
-> policy rule rule5 save
```

For more information about the **configuration snapshot**, **write memory**, and **copy running-config working** commands, see the *OmniSwitch 7700/7800/8800 Switch Management Guide* and the *OmniSwitch CLI Reference Guide.*

For more information about applying rules, see "Applying the Configuration" on page 24-62.

Logging Rules

Logging a rule may be useful for determining the source of firewall attacks. To specify that the switch should log information about flows that match the specified policy rule, use the **policy rule** command with the **log** option. For example:

```
-> policy rule rule5 log
```

To stop the switch from logging information about flows that match a particular rule, use **no** with the **log** keyword. For example:

```
-> policy rule rule5 no log
```

Deleting Rules

To remove a policy rule, use the **no** form of the command.

```
-> no policy rule rule1
```

The rule will be deleted after the next **qos apply**.

Verifying Policy Configuration

To view information about policy rules, conditions, and actions configured on the switch, use the following commands:

show policy condition Displays information about all pending and applied policy conditions or a particular policy condition configured on the switch. Use the **applied** keyword to display information about applied conditions only.

show policy action Displays information about all pending and applied policy actions or a particular policy action configured on the switch. Use the **applied** keyword to display information about applied actions only.

show policy rule Displays information about all pending and applied policy rules or a particular policy rule. Use the **applied** keyword to display information about applied rules only.

show active policy rule Displays applied policy rules that are active (enabled) on the switch.

When the command is used to show output for all pending and applied policy configuration, the following characters may appear in the display:

character	definition
+	Indicates that the policy rule has been modified or has been created since the last **qos apply**.
-	Indicates the policy object is pending deletion.
#	Indicates that the policy object differs between the pending/applied objects.

For example:

```
-> show policy rule
                 Policy          From Prec  Enab Inact Refl  Log  Save
my_rule                          cli     0  Yes   Yes   No   No   Yes
Cnd/Act:                         cond5 -> action2

+my_rule5                        cli     0  Yes  No    No    No   Yes
Cnd/Act:                         cond2 -> pri2

mac1                             cli     0  Yes  No    No    No   Yes
Cnd/Act:                         dmac1 -> pri2
```

The above display indicates that **my_rule** is inactive and is not used to classify traffic on the switch (the Inact field displays **Yes**). The rule **my_rule5** has been configured since the last **qos apply** command was entered, as indicated by the plus (+) sign. The rule will not be used to classify traffic until the next **qos apply**. Only **mac1** is actively being used on the switch to classify traffic.

To display only policy rules that are active (enabled and applied) on the switch, use the **show active policy rule** command. For example:

```
-> show active policy rule

                 Policy           From Prec  Enab Inact Refl  Log  Save    Matches
mac1                              cli  0     Yes  No    No    No  Yes     0
Cnd/Act:                          dmac1 -> pri2
```

In this example, the rule **my_rule** does not display because it is inactive. Rules are inactive if they are administratively disabled through the **policy rule** command, or if the rule cannot be enforced by the current hardware. Although **my_rule5** is administratively active, it is still pending and not yet applied to the configuration. Only **mac1** is displayed here because it is active on the switch.

See the *OmniSwitch CLI Reference Guide* for more information about the output of these commands.

Testing Conditions

Before applying policies to the configuration through the **qos apply** command, you may want to see how the policies will be used to classify traffic. Or you may want to see how theoretical traffic would be classified by policies that are already applied on the switch.

Use the **show policy classify** commands to see how the switch will classify certain condition parameters. This command is used to examine the set of pending policies only. Use the **applied** keyword with the command to examine the applied set of policies only. The command includes a keyword (**l2**, **l3**, **multicast**) to indicate whether the Layer 2, Layer 3, or multicast classifier will be used to classify the traffic.

The keywords used with these commands are similar to the keywords used for the **policy condition** command. The keyword should be relevant to the type of traffic as listed in the table here:

show policy classify l2	**show policy classify l3**	
source port	**source port**	**destination port**
destination port	**destination port**	**destination mac**
source mac	**source ip**	**destination vlan**
destination mac	**destination ip**	**destination interface type**
source vlan	**multicast ip**	**destination ip**
destination vlan	**ip protocol**	
source interface type	**source ip port**	
destination interface type	**destination ip port**	
	source interface type	
	destination interface type	
	tos	
	dscp	

To test a theoretical condition against the set of pending policies, enter the command and the relevant keyword and value. The switch will display information about the potential traffic and attempt to match it to a policy (pending policies only). For example:

```
-> show policy classify l2 destination mac 08:00:20:d1:6e:51
Packet headers:
L2:
  *Port          :                 0/0    ->    0/0
  *IfType        :                 any    ->    any
  *MAC           :       000000:000000    ->    080020:D1E51
  *VLAN          :                   0    ->    0
  *802.1p        : 0
L3/L4:
  *IP            :             0.0.0.0    ->    0.0.0.0
  *TOS/DSCP      : 0/0

Using pending l2 policies
Classify L2 Destination:
  *Matches rule 'yuba': action pri3 (accept)
Classify L2 Source:
  *No rule matched: (accept)
```

The display shows Layer 2 or Layer 3 information, depending on what kind of traffic you are attempting to classify. In this example, the display indicates that the switch found a rule, **yuba**, to classify destination traffic with the specified Layer 2 information.

To test a theoretical condition against the set of applied policies, enter the command with the **applied** keyword. The switch will display information about the potential traffic and attempt to match it to a policy (applied policies only). For example:

```
-> show policy classify l3 applied source ip 143.209.92.131 destination ip
198.60.82.5

Packet headers:
L2:
 *Port           :                 0/0    ->    0/0
 *IfType         :                 any    ->    any
 *MAC            :       000000:000000    ->    000000:000000
 *VLAN           :                   0    ->    0
 *802.1p         : 0
L3/L4:
 *IP             :      143.209.92.131    ->    198.60.82.5
 *TOS/DSCP       : 0/0

Using applied l3 policies
Classify L3:
 *Matches rule ‘r1’: action a1 (drop)
```

In this example, the display indicates that the switch found an applied rule, **r1**, to classify Layer 3 flows with the specified source and destination addresses.

To activate any policy rules that have not been applied, use the **qos apply** command. To delete rules that have not been applied (and any other QoS configuration not already applied), use the **qos revert** command. See “Applying the Configuration” on page 24-62.

Using Condition Groups in Policies

Condition groups are made up of multiple IP addresses, MAC addresses, services, or ports to which you want to apply the same action or policy rule. Instead of creating a separate condition for each address, etc., create a condition group and associate the group with a condition. Groups are especially useful when configuring filters, or Access Control Lists (ACLs); they reduce the number of conditions and rules that must be entered. For information about setting up ACLs, see Chapter 25, "Configuring ACLs."

Commands used for configuring condition groups include the following:

policy network group
policy service group
policy mac group
policy port group

ACLs

Access Control Lists (ACLs) typically use condition groups in policy conditions to reduce the number of rules required to filter particular types of traffic. For more information about ACLs, see Chapter 25, "Configuring ACLs."

Sample Group Configuration

1 Create the group and group entries. In this example, a network group is created:

```
-> policy network group netgroup1 10.10.5.1 10.10.5.2
```

2 Attach the group to a policy condition. For more information about configuring conditions, see "Creating Policy Conditions" on page 24-39.

```
-> policy condition cond3 source network group netgroup1
```

Note. (*Optional*) Use the **show policy network group** command to display information about the network group. Each type of condition group has a corresponding show command. For example:

```
-> show policy network group
Group Name:                 From   Entries
Switch                      blt    4.0.1.166
                                   10.0.1.166

+netgroup1                  cli    10.10.5.1/255.255.255.0
                                   10.10.5.2/255.255.255.0
```

See the *OmniSwitch CLI Reference Guide* for more information about the output of this display. See "Verifying Condition Group Configuration" on page 24-57 for more information about using **show** commands to display information about condition groups.

3 Attach the condition to a policy rule. (For more information about configuring rules, see "Creating Policy Rules" on page 24-41.) In this example, action **act4** has already been configured. For example:

```
-> policy rule my_rule condition cond3 action act4
```

4 Apply the configuration. See "Applying the Configuration" on page 24-62 for more information about this command.

```
-> qos apply
```

The next sections describe how to create groups in more detail.

Creating Network Groups

Use network policy groups for policies based on IP source or destination address. The policy condition will specify whether the network group is a source network group, destination network group, or multicast network group.

- **Default switch group**—Note that by default the switch contains a network group called **switch** that includes all IP addresses configured for the switch itself. This network group may also be used in policy conditions.
- **ACLs**—Typically network groups are used for Access Control Lists. For more information about ACLs, see Chapter 25, "Configuring ACLs."
- **Address Translation**—In addition to using network groups in policy conditions, you can also use network groups in policy actions to translate IP addresses. For an example of address translation, see "Network Address Translation" on page 24-70.

To create a network policy group, use the **policy network group** command. Specify the name of the group and the IP address(es) to be included in the group. Each IP address should be separated by a space. A mask may also be specified for an address. If a mask is not specified, the address is assumed to be a host address.

Note. Network group configuration is not active until the **qos apply** command is entered.

In this example, a policy network group called **netgroup2** is created with two IP addresses. No mask is specified, so the IP addresses are assumed to be host addresses.

```
-> policy network group netgroup2 10.10.5.1 10.10.5.2
```

In the next example, a policy network group called **netgroup3** is created with two IP addresses. The first address also specifies a mask.

```
-> policy network group netgroup3 173.21.4.39 mask 255.255.255.0 10.10.5.3
```

In this example, the 173.201.4.39 address is subnetted, so that any address in the subnet will be included in the network group. For the second address, 10.10.5.3, a mask is not specified; the address is assumed to be a host address.

The network group may then be associated with a condition through the **policy condition** command. The network group must be specified as a **source network group** or **destination network group**. In this example, **netgroup3** is configured for condition **c4** as source network group:

```
-> policy condition c4 source network group netgroup3
```

Note. Network groups may also be included as part of a policy action when you are creating policies for IP address translation. For more information about address translation, see "Network Address Translation" on page 24-70.

To remove addresses from a network group, use **no** and the relevant address(es). For example:

```
-> policy network group netgroup3 no 173.21.4.39
```

This command deletes the 173.21.4.39 address from **netgroup3** after the next **qos apply**.

To remove a network group from the configuration, use the **no** form of the **policy network group** command with the relevant network group name. The network group must not be associated with any policy condition or action. For example:

```
-> no policy network group netgroup3
```

If the network group is not currently associated with any condition or action, the network group **netgroup3** is deleted from the configuration after the next **qos apply**.

If a condition or an action is using **netgroup3**, the switch will display an error message similar to the following:

```
ERROR: netgroup3 is being used by condition 'c4'
```

In this case, remove the network group from the condition first, then enter the **no** form of the **policy network group** command. For example:

```
-> policy condition c4 no source network group
-> no policy network group netgroup3
```

The **policy condition** command removes the network group from the condition. (See "Creating Policy Conditions" on page 24-39 for more information about configuring policy conditions.) The network group will be deleted at the next **qos apply**.

Creating Services

Policy services are made up of TCP or UDP ports or port ranges. They include source or destination ports, or both, but the ports must be the same type (TCP *or* UDP). Mixed port types cannot be included in the same service.

Policy services may be associated with policy service groups, which are then associated with policy conditions; or they may be directly associated with policy conditions.

To create a service, use the **policy service** command. With this command, there are two different methods for configuring a service. You can specify the protocol and the IP port; or you can use shortcut keywords. The following table lists the keyword combinations:

Procedure	Keywords	Notes
Basic procedure for either TCP or UDP service	**protocol** **source ip port** **destination ip port**	*The protocol must be specified with at least one source or destination port.*
Shortcut for TCP service	**source tcp port** **destination tcp port**	*Keywords may be used in combination.*
Shortcut for UDP service	**source udp port** **destination udp port**	*Keywords may be used in combination.*

An IP protocol (TCP or UDP), source IP port and/or destination IP port (or port range) must be associated with a service. IP port numbers are well-known port numbers defined by the IANA. For example, port numbers for FTP are 20 and 21; Telnet is 23.

In this example, a policy service called **telnet1** is created with the TCP protocol number (**6**) and the well-known Telnet destination port number (**23**).

```
-> policy service telnet1 protocol 6 destination ip port 23
```

A shortcut for this command replaces the **protocol** and **destination ip port** keywords with **destination tcp port**:

```
-> policy service telnet1 destination tcp port 23
```

In the next example, a policy service called **ftp2** is created with port numbers for FTP (20 and 21):

```
-> policy service ftp2 protocol 6 source ip port 20-21 destination ip port 20
```

A shortcut for this command replaces the **protocol**, **source ip port**, and **destination ip port** keywords with **source tcp port** and **destination tcp port**:

```
-> policy service ftp2 source tcp port 20-21 destination tcp port 20
```

Multiple services created through the **policy service** command may be associated with a policy service group; or, individual services may be configured for a policy condition. If you have multiple services to associate with a condition, configure a service group and attach it to a condition. Service groups are described in "Creating Service Groups" on page 24-53.

Note. Service configuration is not active until the **qos apply** command is entered.

To remove a policy service, enter the **no** form of the command.

```
-> no policy service ftp2
```

The **ftp2** service is deleted from the configuration at the next **qos apply** if the service is not currently associated with a policy condition or a service group.

Creating Service Groups

Service groups are made up of policy services. First configure the policy service, then create the service group which includes the policy service(s).

Use the **policy service group** command. For example:

```
-> policy service group serv_group telnet1 ftp2
```

In this example, a policy service group called **serv_group** is created with two policy services (**telnet1** and **ftp2**). The policy services were created with the **policy service** command. (See "Creating Services" on page 24-52 for information about configuring policy services.)

Note. The policy service group can include only services with all source ports, all destination ports, or all source and destination ports. For example, the group cannot include a service that specifies a source port and another service that specifies a destination port.

The service group may then be associated with a condition through the **policy condition** command. For example:

```
-> policy condition c6 service group serv_group
```

This command configures a condition called **c6** with service group **serv_group**. All of the services specified in the service group will be included in the condition. (For more information about configuring conditions, see "Creating Policy Conditions" on page 24-39.)

Note. Service group configuration must be specifically applied to the configuration with the **qos apply** command.

To delete a service from the service group, use **no** with the relevant service name. For example:

```
-> policy service group serv_group no telnet1
```

In this example, the service **telnet1** is removed from policy service group **serv_group**.

To delete a service group from the configuration, use the **no** form of the **policy service group** command. The service group must not be associated with any condition. For example:

```
-> no policy service group serv_group
```

Service group **serv_group** will be deleted at the next **qos apply**. If **serv_group** is associated with a policy condition, an error message will display instead. For example:

```
ERROR: serv_group is being used by condition 'c6'
```

In this case, remove the service group from the condition first; then enter the **no policy service group** command. For example:

```
-> policy condition c6 no service group
-> no policy service group serv_group
```

The **policy condition** command removes the service group from the policy condition. (See "Creating Policy Conditions" on page 24-39 for more information about configuring policy conditions.) The service group will be deleted at the next **qos apply**.

Creating MAC Groups

MAC groups are made up of multiple MAC addresses that you want to attach to a condition.

To create a MAC group, use the **policy mac group** command.

For example:

```
-> policy mac group macgrp2 08:00:20:00:00:00 mask ff:ff:ff:00:00:00
00:20:DA:05:f6:23
```

This command creates MAC group **macgrp2** with two MAC addresses. The first address includes a MAC address mask, so that any MAC address starting with 08:00:20 will be included in **macgrp2**.

The MAC group may be then be associated with a condition through the **policy condition** command. Note that the policy condition specifies whether the group should be used for *source* or *destination*. For example:

```
-> policy condition cond3 source mac group macgrp2
```

This command creates a condition called **cond3** that may be used in a policy rule to classify traffic by source MAC addresses. The MAC addresses are specified in the MAC group. For more information about configuring conditions, see "Creating Policy Conditions" on page 24-39.

Note. MAC group configuration is not active until the **qos apply** command is entered.

To delete addresses from a MAC group, use **no** and the relevant address(es):

```
-> policy mac group macgrp2 no 08:00:20:00:00:00
```

This command specifies that MAC address 08:00:20:00:00:00 will be deleted from **macgrp2** at the next **qos apply**.

To delete a MAC group, use the **no** form of the **policy mac group** command with the relevant MAC group name. The group must not be associated with any policy condition. For example:

```
-> no policy mac group macgrp2
```

MAC group **macgrp2** will be deleted at the next **qos apply**. If **macgrp2** is associated with a policy condition, an error message will display instead:

```
ERROR: macgrp2 is being used by condition 'cond3'
```

In this case, remove the MAC group from the condition first; then enter the **no policy mac group** command. For example:

```
-> policy condition cond3 no source mac group
-> no policy mac group macgrp2
```

The **policy condition** command removes the MAC group from the condition. See "Creating Policy Conditions" on page 24-39 for more information about configuring policy conditions. The MAC group will be deleted at the next **qos apply**.

Creating Port Groups

Port groups are made up of slot and port number combinations. Note that there are many built-in port groups, one for each slot on the switch. On the OmniSwitch 8800, the built-in port groups are subdivided by slice. The built in groups are named by slot (**Slot01**, **Slot02**, etc.). To view the built-in groups, use the **show policy port group** command.

To create a port group, use the **policy port group** command. For example:

```
-> policy port group techpubs 2/1 3/1 3/2 3/3
```

The port group may then be associated with a condition through the **policy condition** command. Note that the policy condition specifies whether the group should be used for *source* or *destination*. For example:

```
-> policy condition cond4 source port group techpubs
```

This command creates a condition called **cond4** that may be used in a policy rule to classify traffic by source port number. The port numbers are specified in the port group. For more information about configuring conditions, see "Creating Policy Conditions" on page 24-39.

Note. Port group configuration is not active until the **qos apply** command is entered.

To delete ports from a port group, use **no** and the relevant port number(s).

```
-> policy port group techpubs no 2/1
```

This command specifies that port 2/1 will be deleted from the **techpubs** port group at the next **qos apply**.

To delete a port group, use the **no** form of the **policy port group** command with the relevant port group name. The port group must not be associated with any policy condition. For example:

```
-> no policy port group techpubs
```

The port group **techpubs** will be deleted at the next **qos apply**. If **techpubs** is associated with a policy condition, an error message will display instead:

```
ERROR: techpubs is being used by condition 'cond4'
```

In this case, remove the port group from the condition first; then enter the **no policy port group** command. For example:

```
-> policy condition cond4 no source port group
-> no policy port group techpubs
```

The **policy condition** command removes the port group from the policy condition. (See "Creating Policy Conditions" on page 24-39 for more information about configuring policy conditions.) The port group will be deleted at the next **qos apply**.

Port Groups and Maximum Bandwidth

On the OmniSwitch 7700/7800/8800, if a policy is configured with a port group in the condition and a policy action with maximum bandwidth, the maximum bandwidth is allowed on each active port in the port group. (This functionality is different from the OmniSwitch 6624/6648, which distributes the maximum bandwidth allowance among the active ports in the port group.)

In the following example, a port group (**pgroup**) is created with two ports and attached to a policy condition (**Ports**). A policy action with maximum bandwidth is created (**MaxBw**). The policy condition and policy action are combined in a policy rule called **PortRule**.

```
-> policy port group pgroup 4/1-2
-> policy condition Ports source port group pgroup
-> policy action MaxBw maximum bandwidth 100k
-> policy rule PortRule condition Ports action MaxBw
```

In this example, 100 Kbps will be allowed over each port.

Verifying Condition Group Configuration

To display information about condition groups, use the following **show** commands:

show policy network group Displays information about all pending and applied policy network groups or a particular network group. Use the **applied** keyword to display information about applied groups only.

show policy service Displays information about all pending and applied policy services or a particular policy service configured on the switch. Use the **applied** keyword to display information about applied services only.

show policy service group Displays information about all pending and applied policy service groups or a particular service group. Use the **applied** keyword to display information about applied groups only.

show policy mac group Displays information about all pending and applied MAC groups or a particular policy MAC group configured on the switch. Use the **applied** keyword to display information about applied groups only.

show policy port group Displays information about all pending and applied policy port groups or a particular port group. Use the **applied** keyword to display information about applied groups only.

See the *OmniSwitch CLI Reference Guide* for more information about the syntax and output for these commands.

When the command is used to show output for all pending and applied condition groups, the following characters may appear in the display:

character	definition
+	Indicates that the policy rule has been modified or has been created since the last **qos apply**.
-	Indicates the policy object is pending deletion.
#	Indicates that the policy object differs between the pending/applied objects.

In the example shown here, **netgroup1** is a new network group that has not yet been applied to the configuration.

```
-> show policy network group
Group Name:                 From   Entries
Switch                      blt    4.0.1.166
                                   10.0.1.166
                                   143.209.92.166
                                   192.85.3.1

+netgroup1                  cli    143.209.92.0/255.255.255.0
                                   172.28.5.0/255/255/255.0
```

When the **qos apply** command is entered, the plus sign (+) will be removed from **netgroup1** in the display. See "Applying the Configuration" on page 24-62 for more information about the **qos apply** command.

Using Map Groups

Map groups are used to map 802.1p, ToS, or DSCP values to different values.

Note. Map groups are associated with a policy *action*.

Commands used for creating map groups include the following:

policy map group
policy action map

Sample Map Group Configuration

1 Create the map group with mapping values. For detailed information about map groups and how to set them up, see "How Map Groups Work" on page 24-59 and "Creating Map Groups" on page 24-59.

```
-> policy map group tosGroup 1-2:5 4:5 5-6:7
```

2 Attach the map group to a policy action. See "Creating Policy Actions" on page 24-40 for more information about creating policy actions.

```
-> policy action tosMap map tos to tos using tosGroup
```

Note. (*Optional*) Use the **show policy map group** command to verify the map group.

```
-> show policy map group
Group Name                      From  Entries
+tosGroup                        cli  1-2:5
                                      4:5
                                      5-6:7
```

For more information about this command, see "Verifying Map Group Configuration" on page 24-60 and the *OmniSwitch CLI Reference Guide*.

3 Attach the action to a policy rule. In this example, the condition **Traffic** is already configured. For more information about configuring rules, see "Creating Policy Rules" on page 24-41.

```
-> policy rule r3 condition Traffic action tosMap
```

4 Apply the configuration. For more information about this command, see "Applying the Configuration" on page 24-62.

```
-> qos apply
```

How Map Groups Work

When mapping to the same type of value (802.1p to 802.1p, ToS to ToS, or DSCP to DSCP), the action will result in remapping the specified values. Any values that are not specified in the map group are preserved. In this example, a map group is created for 802.1p bits.

```
-> policy map group Group2 1-2:5 4:5 5-6:7
-> policy action Map1 map 802.1p to 802.1p using Group2
```

The *to* and *from* values are separated by a colon (**:**). If traffic with 802.1p bits comes into the switch and matches a policy that specifies the **Map1** action, the bits will be remapped according to **Group2**. If the incoming 802.1p value is 1 or 2, the value will be mapped to 5. If the incoming 802.1p value is 3, the outgoing value will be 3 (the map group does not specify any mapping for a value of 3). If the incoming 802.1p value is 4, the value will be mapped to 5. If the incoming 802.1p value is 5 or 6, the value will be mapped to 7.

When mapping to a different type of value, however (802.1p to ToS, 802.1p to DSCP), any values in the incoming flow that matches the rule but that are not included in the map group will be zeroed out. For example, the following action specifies the same map group but instead specifies mapping 802.1p to ToS:

```
-> policy action Map2 map 802.1p to tos using Group2
```

In this case, if 802.1p traffic comes into the switch and matches a policy that specifies the **Map2** action, the 802.1p value will be mapped according to **Group2** if the value is part of **Group2**. If the incoming 802.1p value is 2, the value will be mapped to 5; however, if the incoming value is 3, the switch will map the value to zero because there is no mapping in **Group2** for a value of 3.

Note. Ports on which the flow is mapped must be a trusted port; otherwise the flow will not be prioritized with the map value.

Creating Map Groups

To create a map group, use the **policy action map** command. For example, to create a map group called **tosGroup**, enter:

```
-> policy map group tosGroup 1-2:5 4:5 5-6:7
```

The to and from values are separated by a colon (**:**). For example, a value of 2 will be mapped to 5.

Note. Map group configuration is not active until the **qos apply** command is entered.

The remapping group may then be associated with a rule through the **policy action** command. In this example, a policy condition called **Traffic** has already been configured.

```
-> policy action tosMap map tos to tos using tosGroup
-> policy rule r3 condition Traffic action tosMap
```

To delete mapping values from a group, use **no** and the relevant values:

```
-> policy map group tosGroup no 1-2:4
```

The specified values will be deleted from the map group at the next **qos apply**.

To delete a map group, use the **no** form of the **policy map group** command. The map group must not be associated with a policy action. For example:

```
-> no policy map group tosGroup
```

If **tosGroup** is currently associated with an action, an error message similar to the following will display:

```
ERROR: tosGroup is being used by action 'tosMap'
```

In this case, remove the map group from the action, then enter the **no policy map group** command:

```
-> policy action tosMap no map group
-> no policy map group tosGroup
```

The map group will be deleted at the next **qos apply**.

About DSCP Map Groups

In the current software release, when map groups are configured for mapping to or from DSCP values, only one map in each direction may be active on the switch at any one time. For example, you could have the following two actions (**dscpMap** and **tosMap**) being applied to traffic matching conditions that use these actions:

```
-> policy action dscpMap dscp to tos using tosGroup
-> policy action tosMap tos to dscp using dscpGroup
```

The first action (**dscpMap**) maps DSCP to ToS; the second action (**tosMap**) maps ToS to DSCP. When the switch is actively applying these actions to flows (that is, traffic is arriving on the switch that matches policies configured with these actions), any other DSCP map action that is configured cannot be applied to incoming flows.

Note. For Layer 2 flows, you cannot have more than one action that maps DSCP.

Verifying Map Group Configuration

To display information about all map groups, including all pending and applied map groups, use the **show policy map group** command. To display only information about applied map groups, use the **applied** keyword with the command. For more information about the output of this command, see the *OmniSwitch CLI Reference Guide*.

When the command is used to show output for all pending and applied condition groups, the following characters may appear in the display:

character	definition
+	Indicates that the policy rule has been modified or has been created since the last **qos apply**.
-	Indicates the policy object is pending deletion.
#	Indicates that the policy object differs between the pending/applied objects.

In the example here, a new map group, **tosGroup**, has not yet been applied to the configuration.

```
-> show policy map group
Group Name                    From  Entries
+tosGroup                      cli  1-2:5
                                    4:5
                                    5-6:7
```

When the **qos apply** command is entered, the plus sign (+) will be removed from **tosGroup** in the display. See "Applying the Configuration" on page 24-62 for more information about the **qos apply** command.

Applying the Configuration

Configuration for policy rules and many global QoS parameters must specifically be applied to the configuration with the **qos apply** command. Any parameters configured without this command are maintained for the current session but are not yet activated. For example, if you configure a new policy rule through the **policy rule** command, the switch cannot use it to classify traffic and enforce the policy action until the **qos apply** command is entered. For example:

```
-> policy rule my_rule condition c4 action a5
-> qos apply
```

The **qos apply** command must be included in an ASCII text configuration file when QoS commands are included. The command should be included after the last QoS command.

When the configuration is not yet applied, it is referred to as the *pending configuration*.

Global Commands. Many global QoS commands are active immediately on the switch *without* **qos apply**. *The settings configured by these commands will be active immediately.* Other global commands must specifically be applied. The commands are listed in the following table:

Global Commands That Take Effect Immediately	Global Commands That Must Be Applied
qos	**qos default queues**
qos forward log	**qos default bridged disposition**
qos log console	**qos default routed disposition**
qos log lines	**qos default multicast disposition**
qos log level	**qos flow timeout**
debug qos	**qos reflexive timeout**
qos trust ports	**qos fragment timeout**
qos stats interval	**qos nat timeout**
qos revert	**qos classify fragments**
qos flush	**qos classifyl3 bridged**
qos reset	

Port and Policy Commands. All port parameters and policy parameters must be applied with the **qos apply** command.

Port and Policy Commands	
qos port	**policy service**
policy condition	**policy service group**
policy action	**policy mac group**
policy rule	**policy port group**
policy network group	**policy map group**

The pending configuration is useful for reviewing policy rules before actually applying them to the switch. The **show policy classify** commands may be used to review information about new conditions before they are applied on the switch. See “Testing Conditions” on page 24-47.

Applied policy rules may also be administratively disabled (inactive). If a rule is administratively disabled, the rule will exist in the applied configuration but will not be used to classify flows. For more information about disabling/re-enabling a policy rule, see “Creating Policy Rules” on page 24-41.

Deleting the Pending Configuration

Policy settings that have been configured but not applied through the **qos apply** command may be returned to the last applied settings through the **qos revert** command. For example:

```
-> qos revert
```

This command ignores any pending policies (any additions, modifications, or deletions to the policy configuration since the last **qos apply)** and writes the last applied policies to the pending configuration. At this point, the pending policies are the same as the last applied policies.

In this example, there are two new pending policies and three applied policies:

Pending Policies	Applied Policies
rule5	rule1
rule6	rule2
	rule3

If you enter **qos revert**, the configuration will then look like:

Pending Policies	Applied Policies
rule1	rule1
rule2	rule2
rule3	rule3

Flushing the Configuration

In some cases, you may want to remove all of your rules and start over again. To completely erase pending policies from the configuration, use the **qos flush** command. For example:

```
-> qos flush
```

If you then enter **qos apply**, all policy information will be deleted.

In this example, there are two new pending policies and three applied policies:

Pending Policies	Applied Policies
rule5	rule1
rule6	rule2
	rule3

If you enter **qos flush**, the configuration will then look like:

Pending Policies	Applied Policies
	rule1
	rule2
	rule3

In this scenario, you can do one of two things. To write the applied policies back to the pending configuration, use **qos revert**. Or, to delete all policy rule configuration, enter **qos apply**. If **qos apply** is entered, the empty set of pending policies will be written to the applied policies and all policy rule configuration will be deleted.

Interaction With LDAP Policies

The **qos apply**, **qos revert**, and **qos flush** commands do not affect policies created through the PolicyView application. Separate commands are used for loading and flushing LDAP policies on the switch. See Chapter 20, “Managing Authentication Servers,” for information about managing LDAP policies.

Verifying the Applied Policy Configuration

The policy **show** commands have an optional keyword (**applied**) to display only applied policy objects. These commands include:

show policy condition Displays information about all pending and applied policy conditions or a particular policy condition configured on the switch. Use the **applied** keyword to display information about applied conditions only.

show policy action Displays information about all pending and applied policy actions or a particular policy action configured on the switch. Use the **applied** keyword to display information about applied actions only.

show policy rule Displays information about all pending and applied policy rules or a particular policy rule. Use the **applied** keyword to display information about applied rules only.

show policy network group Displays information about all pending and applied policy network groups or a particular network group. Use the **applied** keyword to display information about applied groups only.

show policy service Displays information about all pending and applied policy services or a particular policy service configured on the switch. Use the **applied** keyword to display information about applied services only.

show policy service group Displays information about all pending and applied policy service groups or a particular service group. Use the **applied** keyword to display information about applied groups only.

show policy mac group Displays information about all pending and applied MAC groups or a particular policy MAC group configured on the switch. Use the **applied** keyword to display information about applied groups only.

show policy port group Displays information about all pending and applied policy port groups or a particular port group. Use the **applied** keyword to display information about applied groups only.

show policy map group Displays information about all pending and applied policy map groups or a particular map group. Use the **applied** keyword to display information about applied groups only.

show policy classify Sends Layer 2, Layer 3, or multicast information to the classifier to see how the switch will handle the packet. Use the **applied** keyword to examine only applied conditions.

For more information about these commands, see the *OmniSwitch CLI Reference Guide*.

Policy Applications

Policies are used to classify incoming flows and treat the relevant outgoing flows. There are many ways to classify the traffic and many ways to apply QoS parameters to the traffic.

Classifying traffic may be as simple as identifying a Layer 2 or Layer 3 address of an incoming flow. Treating the traffic might involve prioritizing the traffic, rewriting an IP address, or putting the flow in a server load balancing group. How the traffic is treated (the *action* in the policy rule) typically defines the type of policy:

Type of Policy	Description	Action Parameters Used
Basic QoS policies	Prioritizes particular flows, and/or shapes the bandwidth for the flow	**maximum bandwidth** **minimum bandwidth** **priority** **shared** **maximum depth** **maximum buffers**
ICMP policies	Filters, prioritizes, and/or rate limits ICMP traffic	**disposition** **priority** **maximum bandwidth**
802.1p, ToS, and DSCP tagging or mapping policies	Sets or resets the egress 802.1p, ToS, or DSCP values	**802.1p** **tos** **dscp** **map group**
Network Address Translation (NAT) policies	Translates source or destination IP addresses	**source rewrite address** **destination rewrite address**
Policy Based Routing (PBR)	Redirects routed traffic	**permanent ip** **alternate gateway ip**
Server Load Balancing (SLB) policies	Distributes traffic across multiple servers using a virtual IP address	**load balance group**
Access Control Lists (ACLs)	Groups of policies rules used for filtering traffic (allow/deny)	**disposition**

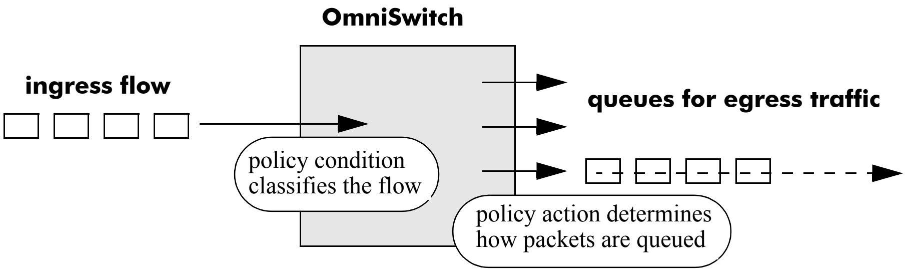
This section describes how to configure basic QoS policies, NAT policies, mapping policies, and routing policies. It provides some information about Server Load Balancing policies; more details about SLB policies are given in Chapter 27, "Configuring Server Load Balancing."

Policies used for Layer 2 and Layer 3/4 filters, commonly refered to as Access Control Lists (ACLs). Filtering, are discussed in Chapter 25, "Configuring ACLs."

Policies may also be used for prioritizing traffic in dynamic link aggregation groups. For more information about dynamic link aggregates, see Chapter 12, "Configuring Dynamic Link Aggregation."

Basic QoS Policies

Traffic priorization and bandwidth shaping may be the most common types of QoS policies. For these policies, any condition may be created; the policy action indicates how the traffic should be prioritized or how the bandwidth should be shaped.


OmniSwitch
ingress flow
queues for egress traffic
policy condition classifies the flow
policy action determines how packets are queued

Note. If multiple addresses, services, or ports should be given the same priority, use a policy condition group to specify the group and associate the group with the condition. See "Using Condition Groups in Policies" on page 24-49 for more information about groups.

Note that some condition parameters may be used in combination only under particular circumstances; also, there are restrictions on condition/action parameter combinations. See "Using Condition Groups in Policies" on page 24-49 and "Condition Combinations" on page 24-6.

Basic Commands

The following **policy action** commands are used for traffic prioritization or shaping:

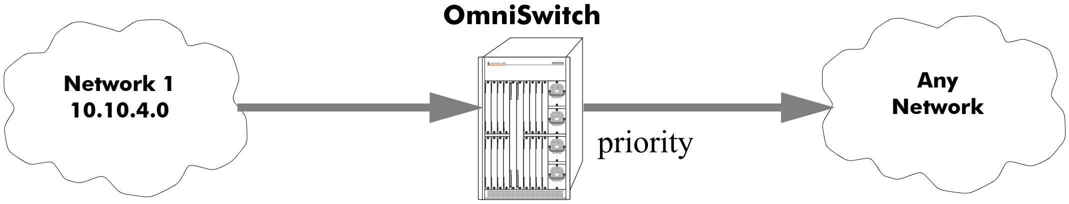
policy action priority
policy action shared
policy action minimum bandwidth
policy action maximum bandwidth
policy action maximum buffers

To set up traffic prioritization and/or bandwidth shaping, follow the steps in the next section. For more information about command syntax and options, see the *OmniSwitch CLI Reference Guide*.

Note that QoS ports may also be configured for bandwidth shaping through the **qos port** commands.

Traffic Prioritization Example

In this example, IP traffic is routed from the 10.10.4.0 network through the OmniSwitch 7800.



To create a policy rule to prioritize the traffic from Network 1, first create a condition for the traffic that you want to prioritize. In this example, the condition is called **ip_traffic**. Then create an action to prioritize the traffic as highest priority. In this example, the action is called **high**. Combine the condition and the action into a policy rule called **rule1**.

```
-> policy condition ip_traffic source ip 10.10.4.0 mask 255.255.255.0
-> policy action high priority 7
-> policy rule rule1 condition ip_traffic action high
```

The rule is not active on the switch until the **qos apply** command is entered on the command line. When the rule is activated, any flows coming into the switch from 10.10.4.0 will be given the highest priority.

Bandwidth Shaping Example

In this example, a specific flow from a source IP address is sent to a queue that will support its maximum bandwidth requirement.

First, create a condition for the traffic. In this example, the condition is called **ip_traffic2**. A policy action (**flowShape**) is then created to enforce a maximum bandwidth requirement for the flow.

```
-> policy condition ip_traffic2 source ip 10.10.5.3
-> policy action flowShape maximum bandwidth 100k
-> policy rule rule2 condition traffic2 action flowShape
```

Note that the bandwidth may be specified in abbreviated units, in this case, **100k**.

The rule is not active on the switch until the **qos apply** command is entered. When the rule is activated, any flows coming into the switch from source IP address 10.10.5.3 will be queued with no more than 100k of bandwidth.

ICMP Policy Example

Policies may be configured for ICMP on a global basis on the switch; ICMP policies may be used for security (for example, to drop traffic from the ICMP blaster virus).

In the following example, a condition called **icmpCondition** is created with no other condition parameters:

```
-> policy condition icmpCondition ip protocol 1
-> policy action icmpAction disposition deny
-> policy rule icmpRule condition icmpCondition action icmpAction
```

This policy (**icmpRule**) drops all ICMP traffic. To limit the dropped traffic to ICMP echo requests (pings), use the **debug qos internal** command with the **pingonly** keyword.

```
-> debug qos internal pingonly
```

The switch will now drop only ICMP echo requests and echo replies. (This functionality is different from the OmniSwitch 6624/6648, which will drop only ICMP echo requests.)

802.1p and ToS/DSCP Marking and Mapping

802.1p values may be mapped to different 802.lp values on an individual basis or by using a map group. In addition, the values may be mapped to ToS or DSCP values instead. Likewise, ToS or DSCP values may be mapped to 802.1p, ToS, or DSCP values on a case-by-case basis or via a map group.

Marking is accomplished with the following commands:

policy action 802.1p
policy action tos
policy action dscp

Mapping is accomplished through the following commands:

policy map group
policy action map

Note the following:

- Priority for the flow is based on the policy action. The value specified for 802.1p, ToS, DSCP, or the map group will determine how the flow is queued.

- The port on which the flow arrives (the ingress port) must be a trusted port. For more information about trusted ports, see "Trusted and Untrusted Ports" on page 24-26.

- For Layer 2 flows, you cannot have more than one action that maps DSCP.

In this example, a policy rule (**marking**) is set up to mark flows from 10.10.3.0 with an 802.1p value of 5:

```
-> policy condition my_condition source ip 10.10.3.0 mask 255.255.255.0
-> policy action my_action 802.1p 5
-> policy rule marking condition my_condition action my_action
```

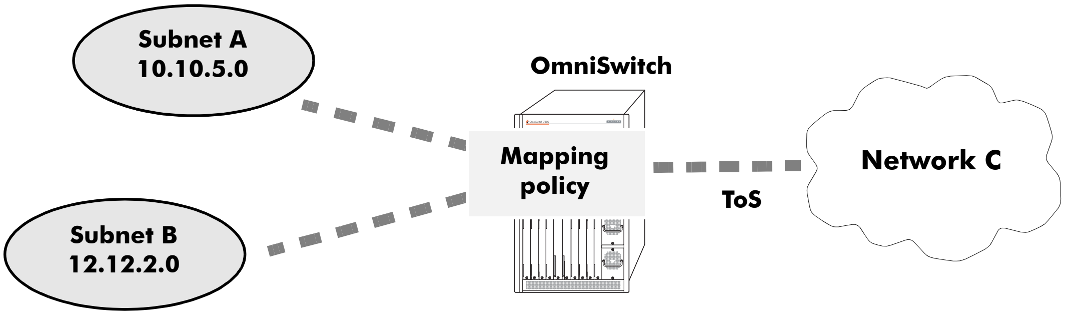
In the next example, the **policy map group** command specifies a group of values that should be mapped; the **policy action map** command specifies what should be mapped (802.1p to 802.1p, 802.1p to ToS, ToS to DSCP, etc.) and the mapping group that should be used. For more details about creating map groups, see "Creating Map Groups" on page 24-59.

Here, two types of 802.1p traffic must be mapped to ToS values into a network called Network C. A map group (**tosGroup**) is created with mapping values.

```
-> policy map group tos_group 1-4:4 5-7:7
-> policy condition SubnetA source ip 10.10.5.0 mask 255.255.255.0
-> policy condition SubnetB source ip 12.12.2.0 mask 255.255.255.0
-> policy action map_action map 802.1p to tos using tos_group
```

The **map_action** specifies that 802.1p values will be mapped to ToS with the values specified in **tos_group**. With these conditions and action set up, two policy rules can be configured for mapping Subnet A and Subnet B to the ToS network:

```
-> policy rule RuleA condition SubnetA action map_action
-> policy rule RuleB condition SubnetB action map_action
```



Mapping Application

Network Address Translation

Network Address Translation (NAT) is used for rewriting a source or destination IP address to another address. A single address may be rewritten, or an entire subnet or list of IP addresses may be rewritten to a group of addresses. These translations would typically be used internally. Translating sets of addresses requires two rules, one for the source and one for the destination.

NAT is used for translating a group of IP addresses to a single address. NAT is typically used for private networks connecting to the Internet. NAT hides the internal addresses from the Internet.

Note. Address translation on the switch translates only addresses in the IP header, not addresses in the packet. Translation is only supported for routed traffic. Bridged traffic will not be translated.

Basic Network Address Translation

To set up IP translation, use a combination of the following CLI commands:

policy condition source ip
policy condition destination ip
policy action source rewrite ip
policy action source rewrite network group
policy action destination rewrite ip
policy action destination rewrite network group
policy rule

Note the following:

- A destination rewrite network group and a destination rewrite IP address cannot be specified in the same action; a source rewrite network group and a source rewrite IP address cannot be specified in the same action.
- IP translation actions cannot be combined with any other type of action (priority, bandwidth shaping, etc.). Source and destination rewrite actions cannot be combined as well.

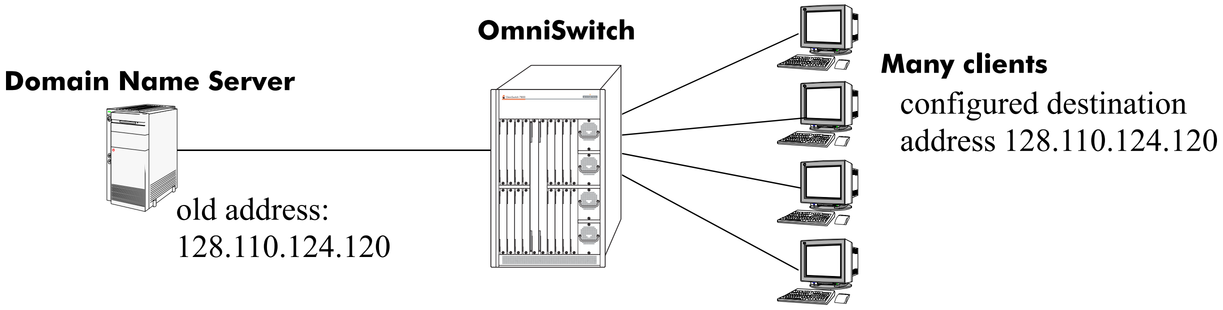
- If you are setting up many-to-many Network Address Translation (NAT), there must be a mechanism in the network for resolving IP addresses in the rewrite group. This may be accomplished with a proxy ARP server located in the network. Alternately, static ARP entries could be configured on the end systems or on the OmniSwitch. (For information about configuring static ARP entries on the OmniSwitch, see Chapter 13, “Configuring IP.”)

IP translation commands are described in the next sections. For more information about additional command options as well as details about syntax, see *OmniSwitch CLI Reference Guide*.

Static Address Translation

Single Address Example

A simple case of static address translation is to rewrite a single IP address to an alternate IP address. For example, if you want to change a server IP address and that server is connected to hundreds of workstations, rather than reconfigure the hundreds of workstations to recognize the new server address, use two static address policies as shown in the example here.


OmniSwitch
Domain Name Server
Many clients
configured destination address 128.110.124.120
old address: 128.110.124.120

1 Create a policy rule (**trans_rule1**) on the switch that will rewrite the destination address.

```
-> policy condition cond4 destination ip 128.110.124.120
-> policy action action2 destination rewrite ip 155.100.39.163
-> policy rule trans_rule1 condition cond4 action action2
```

When traffic arrives on the switch with a destination address of 128.110.124.120, the switch will rewrite the destination address as 155.100.39.163.

2 Create another rule (**trans_rule2**) that will rewrite the source address:

```
-> policy condition cond5 source ip 155.100.39.163
-> policy action action3 source rewrite ip 128.110.124.120
-> policy rule trans_rule2 condition cond5 action action3
```

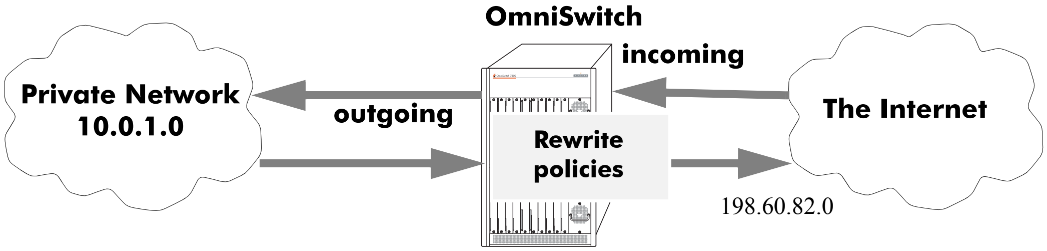
When traffic arrives on the switch with a source IP address of 155.100.39.163, the switch will rewrite the source address as 128.110.124.120.

Make sure to enter the **qos apply** command to activate the rules to the switch. Otherwise they will be saved but not active.

Subnet to Subnet Example

The following example shows how to rewrite a group of IP addresses (a subnet) to another group of addresses. In this example, a private network uses network address 10.0.1.0 and an Internet Service Provider (ISP) assigns the private network the globally-addressable subnet 198.60.82.0. In this scenario, the internal IP addresses, which are not valid Internet addresses, are rewritten to valid Internet addresses.

Note. There must be a one-to-one correspondence between the number of addresses in the policy condition and the number of addresses in the policy action.



Subnet to Subnet (Many-to-Many) NAT

1 First, create a policy rule, called **incoming**, that will rewrite subnet addresses destined for network 198.60.82.0.

```
-> policy condition cond6 destination ip 198.60.82.0 mask 255.255.255.0
-> policy action act6 destination rewrite ip 10.0.1.0 mask 255.255.255.0
-> policy rule incoming cond6 action act6
```

2 Then create another rule, **outgoing**, to rewrite the subnet addresses to another set of addresses.

```
-> policy condition cond7 source ip 10.0.1.0 mask 255.255.255.0
-> policy action act7 source rewrite ip 198.60.82.0 mask 255.255.255.0
-> policy rule outgoing condition cond7 action act7
```

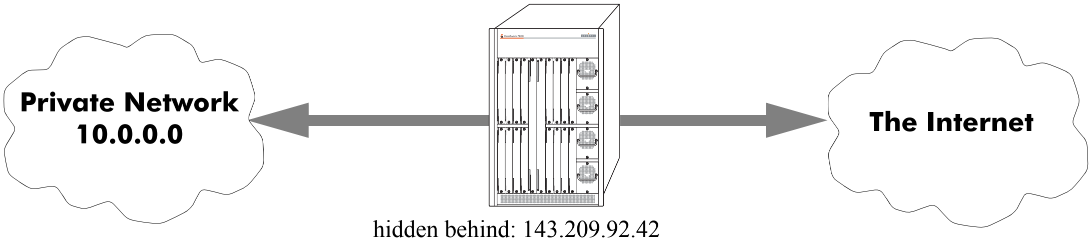
Make sure to enter the **qos apply** command to activate the rules on the switch. Otherwise they will be saved as part of the pending configuration, but will not be active.

Dynamic Address Translation

In addition to static address translation, the switch supports dynamic address translation, where there is a many-to-one address correspondence.

The most common application of this kind of translation is a private network connecting to the Internet through a globally valid IP address. This is useful for hiding the internal addresses from the Internet and is sometimes referred to as *masquerading*.

In this scenario, connections are initiated from the internal to the external network using a private IP source address and a source TCP or UDP port number assigned by the initiating host. The address is rewritten to the IP address provided by the ISP. Packets going from external to internal have their destination addresses and ports changed to the corresponding internal destination address and port. The source ports are assigned dynamically, as connections require it. The source ports distinguish the internal addresses from one another.



Dynamic Address Translation (Many-to-One)

For this type of translation, a single policy rule is required. A network address is given in the condition, and a single address is given in the action. Since the switch multiplexes the various IP addresses from the private network with TCP/UDP ports, it must “remember” the port mappings; no rule is required for the reverse flow.

For example, create a rule called **nat**:

```
-> policy condition internal source ip 10.0.0.0 mask 255.0.0.0
-> policy action external source rewrite ip 143.209.92.42
-> policy rule nat condition internal action external
```

The policy **nat** will rewrite the source address for any traffic from the 10.0.0.0 network to the Internet-friendly address, 143.209.92.42. Note that the mask must be specified for the source address. Traffic destined for the 10.0.0.0 network will be rewritten to the original IP addresses based on the dynamic TCP/UDP port assignment.

Make sure to enter the **qos apply** command to activate the rules on the switch. Otherwise they will be saved as part of the pending configuration, but will not be active.

Server Load Balancing

Server Load Balancing (SLB) is used for balancing traffic destined for a single IP address. An example might be a single IP address connected to multiple Web servers.

Note. Load balancing works only with routed traffic.

When you configure an SLB cluster, a policy condition, action, and rule are automatically configured for the cluster. The switch software names the condition, action, and rule by adding a prefix to the cluster name (**SLB-cond**, **SLB-act**, and **SLB-rule**). Use the **show policy rule** command to display the rule(s).

The condition, action, and rule may be modified, if necessary.

Note. Policies configured for Server Load Balancing take precedence over other policies for traffic matching the specified condition.

The procedure for setting up Server Load Balancing is given in Chapter 27, “Configuring Server Load Balancing.” For more information about Server Load Balancing commands, see the *OmniSwitch CLI Reference Guide*.

Policy Based Routing

Policy Based Routing (PBR) allows a network administrator to define QoS policies that will override the normal routing mechanism for traffic matching the policy condition.

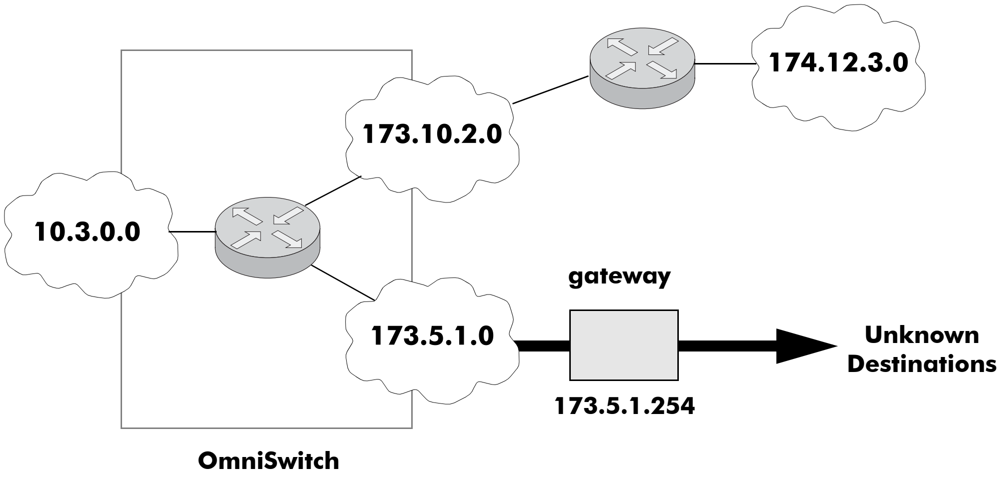
Note. When a PBR QoS rule is applied to the configuration, it is applied to the entire switch, unless you specify a built-in port group in the policy condition.

Policy Based Routing may be used to redirect traffic to a particular gateway based on source or destination IP address, source or destination network group, source or destination TCP/UDP port, a service or service group, IP protocol, or built-in source port group.

Traffic may be redirected to a particular gateway regardless of what routes are listed in the routing table; or traffic may be redirected to a particular gateway only if no route exists in the routing table. Note that the gateway address does not have to be on a directly connected VLAN; the address may be on any network that is learned by the switch.

Note. If the routing table has a default route of 0.0.0.0, traffic matching a PBR policy will be redirected to the route specified in the policy.

For information about viewing the routing table, see Chapter 13, “Configuring IP.”



Routing IP source traffic only for unknown destinations

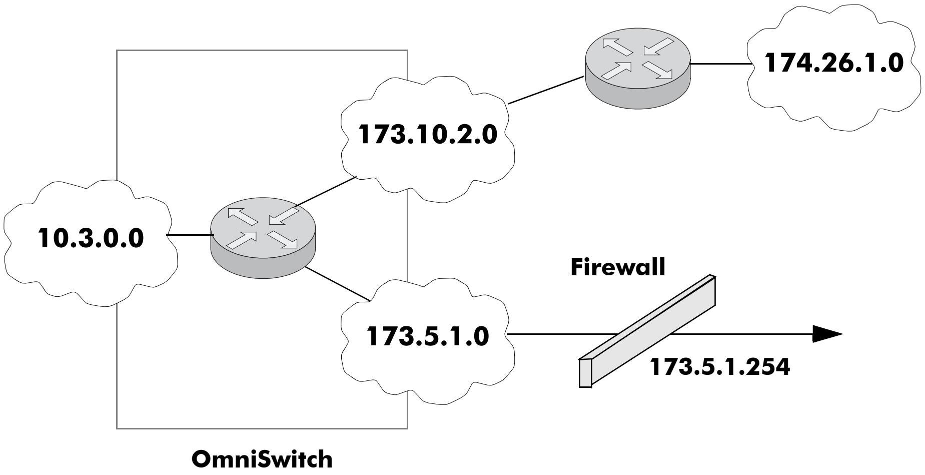
In the above example, routes to 174.12.3.0 are learned by the switch. Traffic originating in 10.3.0.0 and destined for 174.12.3.0 would be routed based on the route in the switch’s routing table. Traffic originating in 10.3.0.0 with an unknown destination, however, would normally trigger the switch to broadcast a route request. In this setup, however, the traffic is automatically forwarded through 173.5.1.254 if the following policy is configured on the switch:

```
-> policy condition Traffic2 source ip 10.3.0.0 mask 255.255.0.0
-> policy action Redirect alternate gateway ip 173.5.1.254
-> policy rule Unknown condition Traffic2 action Redirect
```

In this scenario, any traffic originating in the 10.3 network with an unknown destination will be forwarded through 173.5.1.254.

Make sure to enter the **qos apply** command to activate the policy rule on the switch. Otherwise the rule will be saved as part of the pending configuration, but will not be active.

Policy Based Routing may also be used to redirect untrusted traffic to a firewall. In this case, note that reply packets will be not be allowed back through the firewall.



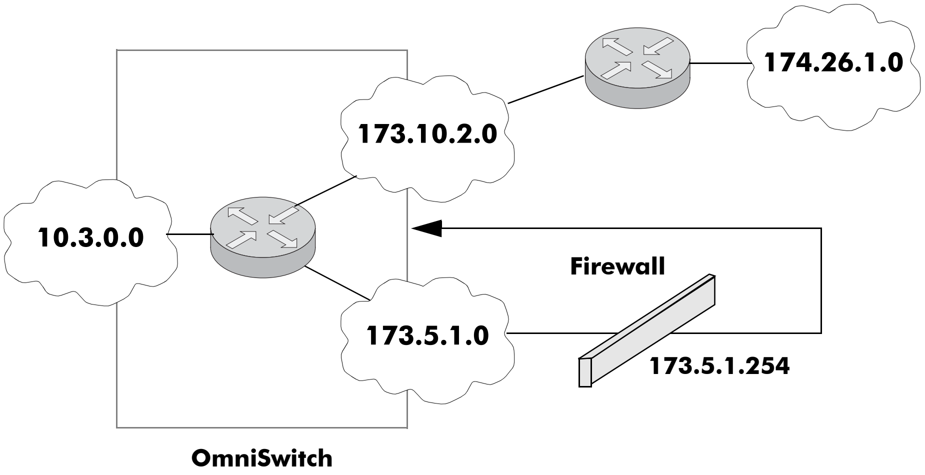
Routing all IP source traffic through a firewall

In this example, all traffic originating in the 10.3 network is routed through the firewall, regardless of whether a route exists.

```
-> policy condition Traffic3 source ip 10.3.0.0 mask 255.255.0.0
-> policy action Firewall permanent gateway ip 173.5.1.254
-> policy rule Redirect_All condition Traffic3 action Firewall
```

Note that the functionality of the firewall is important. In the example, the firewall is sending the traffic to be routed remotely. If you instead set up a firewall to send the traffic back to the switch to be routed, you should set up the policy condition with a built-in source port group so that traffic coming back from the firewall will not get looped and sent back out to the firewall.

For example:



Using a Built-In Port Group

In this scenario, traffic from the firewall is sent back to the switch to be re-routed. But because the traffic re-enters the switch through a port that is not in the Slot01 port group, the traffic does not match the Redirect_All policy and is routed normally through the switch.

```
-> policy condition Traffic3 source ip 10.3.0.0 mask 255.255.0.0 source port
group Slot01
-> policy action Firewall permanent gateway ip 173.5.1.254
-> policy rule Redirect_All condition Traffic3 action Firewall
```

Make sure to enter the **qos apply** command to activate the policy rule on the switch. Otherwise the rule will be saved as part of the pending configuration, but will not be active.

25 Configuring ACLs

Access Control Lists (ACLs) are Quality of Service (QoS) policies used to control whether or not packets are allowed or denied at the switch or router interface. ACLs are sometimes referred to as filtering lists.

ACLs are distinguished by the kind of traffic they filter. In a QoS policy rule, the type of traffic is specified in the policy condition. The policy action determines whether the traffic is allowed or denied. For detailed descriptions about configuring policy rules, see Chapter 24, "Configuring QoS."

In general, the types of ACLs include:

- *Layer 2 ACLs*—for filtering traffic at the MAC layer. Usually uses MAC addresses or MAC groups for filtering.
- *Layer 3/4 ACLs*—for filtering traffic at the network layer. Typically uses IP addresses or IP ports for filtering; note that IPX filtering is not supported.
- *Multicast ACLs*—for filtering IGMP traffic.

In This Chapter

This chapter describes ACLs and how to configure them through the Command Line Interface (CLI). CLI commands are used in the configuration examples; for more details about the syntax of commands, see the *OmniSwitch CLI Reference Guide*.

Configuration procedures described in this chapter include:

- **Setting the Global Disposition**. The disposition specifies the general allow/deny policy on the switch. See "Setting the Global Disposition" on page 25-9.
- **Creating Condition Groups for ACLs**. Groups are used for filtering on multiple addresses, ports, or services. The group is then associated with the policy condition. See "Creating Condition Groups For ACLs" on page 25-11.
- **Creating Policy Rules for ACLs**. Policy rules for ACLs are basically QoS policy rules. Specific parameters for ACLs are described in this chapter. See "Configuring ACLs" on page 25-11.
- **Using ACL Security Features**. A special port group, service group, and policy rule combinations are provided to help improve network security. See "Using ACL Security Features" on page 25-18.

ACL Specifications

These specifications are the same as those for QoS in general:

Maximum number of policy rules	2048
Maximum number of policy conditions	2048
Maximum number of policy actions	2048
Maximum number of policy services	256
Maximum number of groups (network, MAC, service, port)	1024
Maximum number of group entries	512 per group
Maximum number of flows	64000

ACL Defaults

The following table shows the defaults for ACLs:

Parameter	Command	Default
Global bridged disposition	**qos default bridged disposition**	accept
Global routed disposition	**qos default routed disposition**	accept
Global multicast disposition	**qos default multicast disposition**	accept
Policy rule disposition	**policy rule disposition**	accept
Policy rule precedence	**policy rule precedence**	0 (lowest)

Note that in the current software release, the **deny** and **drop** options produce the same effect; that is, that traffic is silently dropped.

For more information about QoS defaults in general, see Chapter 24, “Configuring QoS.”

Quick Steps for Creating ACLs

1 Set the global disposition for bridged, routed, or multicast traffic. By default, all flows that do match any policies are allowed on the switch. Typically, you may want to deny traffic for all Layer 3 flows that come into the switch and do not match a policy, but allow any Layer 2 (bridged) flows that do not match policies. For example:

```
-> qos default routed disposition deny
```

2 Create policy condition groups for multiple addresses or services that you want to filter. (If you have a single address to filter, you can skip this step and simply include the address, service, or port in the policy condition.) An example:

```
-> policy network group NetGroup1 192.68.82.0 mask 255.255.255.0 192.60.83.0
mask 255.255.255.0
```

3 Create a policy condition using the **policy condition** command. If you created a network group, MAC group, service group, or port group, specify the group as part of the condition.

```
-> policy condition Lab3 source network group NetGroup1
```

Note. (*Optional*) Test the condition with the **show policy classify** command using information from the policy condition. For example:

```
-> show policy classify l3 source ip 192.68.82.0
```

This command displays information about whether the indicated parameter may be used to classify traffic based on policies that are configured on the switch. For more information about testing conditions, see "Testing Conditions" on page 24-47 in Chapter 24, "Configuring QoS."

4 Create a policy action with the **policy action** command. Use the keyword **disposition** and indicate whether the flow(s) should be accepted or denied.

```
-> policy action Yes disposition accept
```

5 Create a policy rule with the **policy rule** command and include the relevant condition and action. The keyword **reflexive** is required if there is another policy that would deny the reverse flow. Use the keyword **precedence** to specify the priority of this rule over other rules for traffic matching the specified condition.

```
-> policy rule lab_rule1 condition Lab3 action Yes precedence 65535
```

6 For Layer 3 filtering, make sure that IP router ports are available on the VLANs on which you will be routing. Use the **vlan router ip** command. For example:

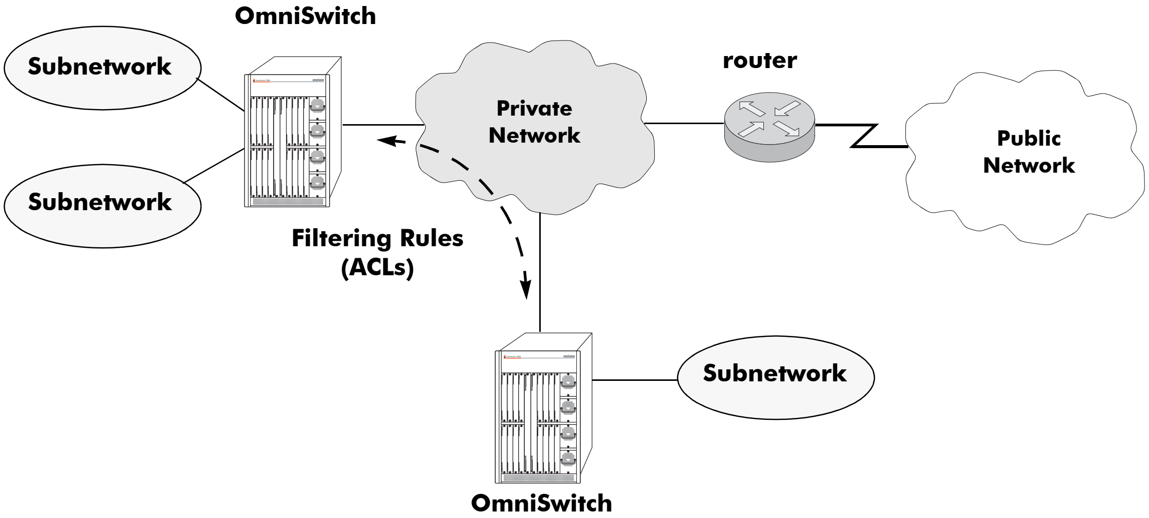
```
-> vlan 2 router ip 192.68.82.1
```

For more information about setting up IP routing, see Chapter 13, "Configuring IP."

7 Apply the policy configuration using the **qos apply** command. For details about using this command, see "Applying the Configuration" on page 24-62 in Chapter 24, "Configuring QoS."

ACL Overview

ACLs provide moderate security between networks. The following illustration shows how ACLs may be used to filter subnetwork traffic through a private network, functioning like an internal firewall for LANs.



Basic ACL Application

When traffic arrives on the switch, the switch checks its policy database to attempt to match Layer 2 or Layer 3/4 information in the protocol header to a filtering policy rule. If a match is found, it applies the relevant *disposition* to the flow. Disposition determines whether a flow is allowed or denied. There is a global disposition (the default is **accept**), and individual rules may be set up with their own dispositions.

Note. In some network situations, it is recommended that the global disposition be set to **deny**, and that rules be created to allow certain types of traffic through the switch. To set the global disposition to deny, use the **qos default bridged disposition**, **qos default routed disposition**, and the **qos default multicast disposition** commands. See "Setting the Global Disposition" on page 25-9 for more information about these commands.

When multiple policy rules exist for a particular flow, the rule with the highest precedence is applied to the traffic. See "Rule Precedence" on page 25-5 for more information about precedence.

Filtering rules are typically reflexive, that is, they require that a reverse flow (such as traffic coming back from a filtered TCP sessions) would be allowed through the switch. For more information about reflexive policies, see "Reflexive Rules" on page 25-7.

Note. QoS policy rules may also be used for traffic prioritization and bandwidth shaping, address translation, source load balancing, and other network scenarios. For a general discussion of QoS policy rules, see Chapter 24, "Configuring QoS."

Rule Precedence

The switch attempts to classify flows coming into the switch according to policy precedence. More than one rule may have the same condition, but the rule with the highest precedence will be applied to the flow matching the condition.

Precedence is determined by the following, in order:

- **The type of QoS rule** (Layer 2 source, Layer 2 destination, or Layer 3)—When a flow comes into the switch, the Layer 2 source rules are examined first for a match. If no match is found, the Layer 2 destination rules are examined. If no match is found, the Layer 3 rules are examined.
- **Precedence value**—Each policy has a precedence value. The value may be user-configured through the **policy rule** command in the range from 0 (lowest) to 65535 (highest). (The range 30000 to 65535 is typically reserved for PolicyView.) By default, a policy rule has a precedence of 0. See the next section for information about configuring precedence for a particular rule.
- **Configured rule order**—If a flow matches more than one rule in a particular precedence list (for example, the Layer 2 source list), and both rules have the same precedence value, the rule that was *configured first* in the list will take precedence.

If a flow matches more than one rule in a particular precedence list (for example, the Layer 2 source list), the precedence determines which rule the switch will apply to the flow.

Note. Policy rules configured for Server Load Balancing (SLB) always take precedence over any other rules configured for traffic matching the same condition. For more information about configuring QoS rules for SLB, see Chapter 27, “Configuring Server Load Balancing.”

Example: Rule Type

In the following example, two rules (**SourceRule** and **DestRule**) are created to deny Layer 2 traffic with action **a1**. Two conditions are set up for the source and destination parameters (**L2source** and **L2dest**).

```
-> policy action a1 disposition deny
-> policy condition L2source source vlan 3
-> policy condition L2dest destination mac 00:20:da:05:f6:23
-> policy rule SourceRule condition L2source action a1
-> policy rule DestRule condition L2dest action a1 precedence 200
```

In this scenario, if traffic comes into the switch on VLAN 3 and is destined for MAC address 00:20:da:05:f6:23, the flow will match **SourceRule**, despite the higher precedence setting configured for **DestRule**. (**SourceRule** is using the default precedence setting, since it is not specified. The default is 100.) Layer 2 source rules always take precedence.

Note. If you enable Layer 2 traffic to be classified as Layer 3 (through the **qos classifyl3 bridged** command), and you have Layer 2 destination rules configured on the switch, make sure Layer 2 destination rules do not match the traffic that you want to classify as Layer 3. Otherwise, the flows will not be classified as Layer 3 since Layer 2 destination rules take precedence over Layer 3 rules.

Example: Precedence Value in Policy Rule

In general, more specific rules (that is, rules with more condition/action parameters) should be given a higher precedence than general rules. This is important for flows coming into the switch that potentially match more than one condition.

In the next example, both Layer 3 conditions, **General** and **Specific**, have the same destination address, but **Specific** also includes a source address.

```
-> policy condition General destination ip 10.0.1.0 mask 255.255.255.0
-> policy condition Specific source ip 10.0.2.3 destination ip 10.0.1.0 mask
255.255.255.0
```

The conditions are then associated with two different policies:

```
-> policy rule r7 precedence 200 condition General action a1
-> policy rule r8 precedence 100 condition Specific action a2
```

In this scenario, if a flow comes in with a destination address of 10.0.1.0 and source address of 10.0.2.3, the flow will match the **General** condition rather than the **Specific** condition since policy rule **r7** has a higher precedence.

To force the switch to classify on the more specific condition, give the lower precedence value to the more general rule:

```
-> policy rule r7 precedence 100 condition General action a1
-> policy rule r8 precedence 200 condition Specific action a2
```

Now if traffic destined for 10.0.1.0 with a source address of 10.0.2.3 comes into the switch, it will match **r8**, the more specific rule.

Note that if both rules are configured with the same precedence, the rule that was configured first will be used. For example:

```
-> policy rule r7 precedence 200 condition General action a1
-> policy rule r8 precedence 200 condition Specific action a2
```

In this scenario, policy rule **r7** is configured first, so incoming traffic with a destination address of 10.0.1.0 and a source address of 10.0.2.3 will match **r7**.

Example: Rule Order

If a policy is configured with the same precedence value as another policy of the same type with the same condition, the policy that was created first has the higher precedence. The new policy is considered lower priority.

In this example, two Layer 3 policy rules are created in the order shown. Both rules have the same precedence value.

```
-> policy rule r1 precedence 100 condition c1 action a1
-> policy rule r2 precedence 100 condition c1 action a2
```

When traffic comes into the switch that matches **c1**, the switch will use rule **r1**.

Reflexive Rules

Reflexive policies allow a reverse flow back through the switch when the reverse flow would normally be denied. For example, a Telnet session in one direction will prompt a reply back from the host. If a policy is created to deny traffic from that host to the switch, the replies from the host will still be accepted on the switch if the Telnet session policy is configured as a reflexive policy.

Note. Reflexive policies are only supported for TCP or UDP traffic. Dynamic port negotiation is not supported.

In the following example, two rules are created. **Rule2** drops traffic destined for the 192.68.82.0 subnet; however, **Rule1** is configured as reflexive, so that a reverse flow going to that subnet will not be dropped.

```
-> policy condition Source1 source ip 192.68.82.0 mask 255.255.255.0
-> policy condition Dest1 destination ip 192.68.82.0 mask 255.255.255.0

-> policy action Permit disposition accept
-> policy action Prevent disposition deny

-> policy rule Rule1 condition Source1 action Permit reflexive
-> policy rule Rule2 condition Dest1 action Prevent
```

In this scenario, TCP or UDP traffic (such as Telnet) from the subnet will receive replies back from the host despite that fact that **Rule2** is configured to drop traffic destined for the subnet.

The switch has a global timeout that specifies the amount of time the switch will wait for reflexive flows. The default is 300 seconds. To change the timeout, use the **qos reflexive timeout** command. For more information about this command, see "Reflexive Timeout" on page 24-20.

Interaction With Other Features

- **IP Routing**—IP routing must be enabled on the switch for Layer 3 ACLs. See Chapter 13, "Configuring IP," for information about setting up routing.
- **Routing Protocols**—Layer 3 filtering is compatible with routing protocols on the switch, including RIP and OSPF. If VRRP is also running, all VRRP routers on the LAN must be configured with the same filtering rules; otherwise the security of the network will be compromised. For more information about VRRP, see Chapter 18, "Configuring VRRP."
- **Bridging**—Layer 2 ACLs are supported for both bridged and routed traffic. Layer 3 ACLs are typically only performed on routed traffic, but the switch may be set to classify Layer 3 information in bridged frames. For information about configuring the switch to classify Layer 3 information in bridged frames, see "Classifying Bridged Traffic as Layer 3" on page 24-22.

Valid Combinations

There are limitations to the types of conditions that may be combined in a single rule. A brief overview of these limitations is listed here:

- Layer 2 and Layer 3/4 conditions should not be combined.
- Layer 2 conditions cannot combine source and destination parameters.
- Type of Service (ToS) and Differentiated Services Code Point (DSCP) values cannot be combined in a single condition.
- Individual items and their correponding groups cannot be combined in the same condition. For example, a source IP address cannot be included in a condition with a source IP network group.

For more information about supported combinations, see "Condition Combinations" on page 24-6 and "Condition/Action Combinations" on page 24-8 in Chapter 24, "Configuring QoS."

ACL Configuration Overview

This section describes the QoS CLI commands used specifically to configure ACLs. ACLs are basically a type of QoS policy, and the commands used to configure ACLs are a subset of the switch's QoS commands. For information about basic configuration of QoS policies, see Chapter 24, "Configuring QoS."

To configure an ACL, the following general steps are required:

1 **Set the global disposition**. This step is described in "Setting the Global Disposition" on page 25-9.

2 **Create a condition for the traffic to be filtered**. This step is described in "Creating Condition Groups For ACLs" on page 25-11 and "Creating Policy Conditions For ACLs" on page 25-12.

3 **Create an action to accept or deny the traffic**. This step is described in "Creating Policy Actions For ACLs" on page 25-12.

4 **Create a policy rule that combines the condition and the action.** This step is described in "Creating Policy Rules for ACLs" on page 25-12.

For a quick tutorial on how to configure ACLs, see "Quick Steps for Creating ACLs" on page 25-3.

Setting the Global Disposition

By default, flows that do not match any policies are accepted on the switch. You may configure the switch to deny any flow that does not match a policy.

Note. Note that the global disposition setting applies to all policy rules on the switch, not just those that are configured for ACLs.

The global commands include:

qos default bridged disposition
qos default routed disposition
qos default multicast disposition

To change the global default dispositions, use these commands with the desired disposition value (**accept**, **drop**, or **deny**).

For Layer 3 ACLs, it is recommended that the global dispositions be set to **deny**. For example, the following command drops any routed traffic coming into the switch that does not match a policy:

```
-> qos default routed disposition deny
```

Policies may then be set up to allowed routed traffic through the switch.

Note that in the current release of the QoS software, the **drop** and **deny** keywords produce the same result (flows are silently dropped; no ICMP message is sent).

For more information about the global disposition commands, see Chapter 24, “Configuring QoS,” and the *OmniSwitch CLI Reference Guide*.

Important. If you set the global bridged disposition (using the **qos default bridged disposition** command) to **deny** or **drop**, it will result in dropping all Layer 2 traffic from the switch that does not match any policy to accept traffic. You must create policies (one for source and one for destination) to allow Layer 2 traffic on the switch.

If you set the bridged disposition to **deny** or **drop**, and you configure Layer 2 ACLs, you will need two rules for each type of filter. For more information, see “Layer 2 ACLs” on page 25-13.

Creating Condition Groups For ACLs

Condition groups for ACLs are made up of multiple IP addresses, MAC addresses, services, or IP ports to which you want to apply the same disposition. Instead of creating a separate condition for each policy rule, create a condition group and associate the group with the condition. This reduces the number of rules you would have to configure (one for each address, service, or port).

The commands used for creating condition groups include:

policy network group
policy mac group
policy service
policy service group
policy port group

For example:

```
-> policy network group netgroup2 10.10.5.1 10.10.5.2 10.10.5.3
-> policy condition cond2 source network group netgroup2
```

This command configures a network group (**netgroup2**) of three IP addresses. The network group is then configured as part of a policy condition (**cond2**). The condition specifies that the addresses in the group are source addresses. (For all condition groups except service groups, the policy condition specifies whether the condition group is a *source* or *destination* group.)

If a network group was not used, a separate condition would have to be created for each IP address. Subsequently, a corresponding rule would have to be created for each condition. Using a network group reduces the number of rules required.

For more details about using groups in policy conditions, see “Using Condition Groups in Policies” on page 24-49 in Chapter 24, “Configuring QoS.”

Configuring ACLs

This section describes in detail the procedures for configuring ACLs. For more information about how to configure policies in general, see Chapter 24, “Configuring QoS.” Command syntax is described in detail in the *OmniSwitch CLI Reference Guide*.

The basic commands for configuring ACL rules are the same as those for configuring policy rules:

policy condition
policy action
policy rule

Creating Policy Conditions For ACLs

A policy condition for IP filtering may include a particular source IP address, destination IP address, source IP port, or destination IP port. Or, the condition may simply refer to the network group, MAC group, port group, or service group. Typically ACLs use group keywords in policy conditions. A single rule, therefore, filters traffic for multiple addresses or ports.

For example:

```
-> policy port group pgroup1 3/1-2 4/3 5/4
-> policy condition c2 source port group pgroup1
```

In this example, a Layer 2 condition (**c2**) specifies that traffic matches the ports included in the **pgroup1** port group. The condition also specifies that the port group is a source group. Any traffic coming in on ports 1 or 2 on slot 3, port 3 on slot 4, or port 4 on slot 5 will match condition **c2**.

For more information about condition groups, see “Creating Condition Groups For ACLs” on page 25-11.

The following table lists the keywords for the **policy condition** command that are typically used for the different types of ACLs:

Layer 2 ACL Condition Keywords	Layer 3/4 ACL Condition Keywords	Multicast ACL Condition Keywords
source mac	**source ip**	**multicast ip**
source mac group	**source network group**	**multicast network group**
destination mac	**destination ip**	**destination ip**
destination mac group	**destination network group**	**destination vlan**
source vlan	**source ip port**	**destination port**
destination vlan	**destination ip port**	**destination port group**
source port	**service**	**destination mac**
source port group	**service group**	**destination mac group**
destination port	**ip protocol**	
destination port group	**destination port**	
source interface type	**destination port group**	
destination interface type	**destination interface type**	

Note that the individual address, service, or port cannot be used in conjunction with the same type of condition group. For example, you cannot specify in the same rule both a source MAC address and a source MAC group.

Creating Policy Actions For ACLs

A policy action for IP filtering specifies a *disposition*, that is, whether the flow is accepted or denied on the switch. To create a policy action, use the **policy action** command. Use the **disposition** keyword to define whether the flow is accepted (**accept**) or denied (**deny**). For example:

```
-> policy action a1 disposition accept
```

If you do not specify a disposition for the policy action, the default (**accept**) will be used.

Creating Policy Rules for ACLs

A policy rule is made up of a condition and an action. The policy rule also specifies if the rule is reflexive, that is, if it applies to reply packets from traffic that matches the condition. If a rule is reflexive, the reply packets will be filtered the same as the initial flow. Typically Layer 3 ACLs are configured to be reflexive.

For example, to create a policy rule for filtering IP addresses, which is a Layer 3 ACL, use the **policy rule** command with the **condition** and **action** keywords. The **precedence** and **reflexive** keywords are optional. By default rules have a precedence of 0. See "Rule Precedence" on page 25-5 for more information about precedence. See "Reflexive Rules" on page 25-7 for more information about making rules reflexive.

```
-> policy condition c3 source ip 10.10.4.8
-> policy action a1 disposition accept
-> policy rule rule7 precedence 65535 condition c3 action a1 reflexive
```

In this example, any traffic matching condition **c3** will match **rule7**; **rule7** is configured with the highest precedence value. If any other Layer 3 rules are configured for traffic with a source address of 10.10.4.8, **rule7** will take precedence over the other rules. (For more information about precedence, see "Rule Precedence" on page 25-5.) The action configured for the rule, **a1**, allows traffic from 10.10.4.8, so the flow will be accepted on the switch.

The rule will not be used to classify traffic or enforce the policy until the **qos apply** command is entered. For information about applying policy parameters, see "Applying the Configuration" on page 24-62 in Chapter 24, "Configuring QoS."

Layer 2 ACLs

Layer 2 filtering filters traffic at the MAC layer. The QoS software works in conjunction with the source learning mechanism in the switch to filter Layer 2 traffic. Layer 2 filtering may be done for both bridged and routed packets. As MAC addresses are learned on the switch, QoS classifies the traffic based on:

- MAC address or MAC group
- VLAN
- Physical slot/port or port group
- Interface type

The switch classifies the MAC address as both source ***and*** destination. The condition parameters in the policy rule *must be all source parameters or all destination parameters*.

The following **policy condition** keywords are used for Layer 2 ACLs:

Layer 2 ACL Condition Keywords	
source mac	**destination mac**
source mac group	**destination mac group**
source vlan	**destination vlan**
source port	**destination port**
source port group	**destination port group**
source interface type	**destination interface type**

A group and an individual item cannot be specified in the same condition. For example, a source MAC address and a source MAC group cannot be specified in the same condition.

Note that some combinations of Layer 2 conditions may not be valid. Refer to "Condition Combinations" on page 24-6 and "Condition/Action Combinations" on page 24-8 in Chapter 24, "Configuring QoS."

If the default bridged disposition is set to **drop** or **deny**, any rules for allowing Layer 2 traffic through the switch must be configured in two instances, once for source and once for destination.

Layer 2 ACL: Example 1

In this example, the default bridged disposition is **accept** (the default). Since the default is **accept**, the **qos default bridged disposition** command would only need to be entered if the disposition had previously been set to **deny**. The command is shown here for completeness.

```
-> qos default bridged disposition accept
-> policy condition Address1 source mac 080020:112233 source vlan 5
-> policy action BlockTraffic disposition deny
-> policy rule FilterA condition Address1 action BlockTraffic
```

In this scenario, traffic with a source MAC address of 08:00:20:11:22:33 coming in on VLAN 5 would match condition **Address1**, which is a condition for a policy rule called **FilterA**. **FilterA** is then applied to the flow. Since **FilterA** has an action (**BlockTraffic**) that is set to deny traffic, the flow would be denied on the switch.

Layer 2 ACL: Example 2

In this example, the default bridged disposition is set to **deny**.

Important. Setting the global bridged disposition to **deny** or **drop** is *not* recommended. This setting effectively drops all Layer 2 traffic on the switch that does not match any **accept** policy. The following example is included to show that you must configure two rules to allow Layer 2 flows in this atypical scenario.

To allow Layer 2 traffic into the switch, two rules must be configured, one for Layer 2 source traffic, and one for Layer 2 destination traffic.

```
-> qos default bridged disposition deny
-> policy condition cond4 source mac 0020da:000000 mask ffffff:000000
-> policy action AllowTraffic disposition accept
-> policy rule Filter1 condition cond4 action AllowTraffic

-> policy condition cond5 destination interface type ethernet
-> policy rule Filter2 condition cond4 action AllowTraffic
```

Since the QoS software classifies the MAC address twice, after **Filter1** is applied to the configuration, the switch will classify any traffic with a MAC address starting with 0020da as both source and destination. Condition **cond4** allows the source traffic on the switch, but the destination traffic will be denied unless another rule is set up. (Note that the source and destination parameters cannot both be specified in the same condition for Layer 2 ACLs.)

In this example, **cond5** is set up for classifying Layer 2 destination traffic on Ethernet interfaces, and **Filter2** is created with **cond5**. Now when Layer 2 flows with a MAC address starting with 0020da arrive on the switch destined for any Ethernet interface, the flows will be allowed on the switch.

Layer 3 ACLs

The QoS software in the switch filters routed traffic at Layer 3. For Layer 3 filters, typically IP routing must be enabled; however, the switch may be configured to filter Layer 3 headers in bridged traffic. Use the **qos classifyl3 bridged** command to filter Layer 3 headers for bridged traffic. For more information, see "Classifying Bridged Traffic as Layer 3" on page 24-22.

For Layer 3 filtering, the QoS software in the switch classifies traffic based on:

- Source IP address or source network group
- Destination IP address or destination network group
- IP protocol
- Source TCP/UDP port
- Destination TCP/UDP port or service or service group
- Destination slot/port or destination port group
- Destination interface type

The following **policy condition** keywords are used for Layer 3 ACLs:

Layer 3/4 ACL Condition Keywords
source ip **source network group** **destination ip** **destination network group** **source ip port** **destination ip port** **service** **service group** **ip protocol** **destination port** **destination port group** **destination interface type**

Layer 3 ACL: Example 1

In this example, the default routed disposition is **accept** (the default). Since the default is **accept**, the **qos default routed disposition** command would only need to be entered if the disposition had previously been set to **deny**. The command is shown here for completeness.

```
-> qos default routed disposition accept
-> policy condition addr2 source ip 192.68.82.0 source ip port 23 ip protocol 6
-> policy action Block disposition deny
-> policy rule FilterL31 condition addr2 action Block
```

Traffic with a source IP address of 192.68.82.0, a source IP port of 23, using protocol 6, will match condition **addr2**, which is part of **FilterL31**. The action for the filter (**Block**) is set to deny traffic. The flow will be dropped on the switch.

Layer 3 ACL: Example 2

This example uses condition groups to combine multiple IP addresses in a single condition. The default disposition is set to **deny**.

```
-> qos default routed disposition deny
-> policy network group GroupA 192.60.22.1 192.60.22.2 192.60.22.0
-> policy condition cond7 destination network group GroupA
-> policy action Ok disposition accept
-> policy rule FilterL32 condition cond7 action Ok
```

In this example, a network group, **GroupA**, is configured with three IP addresses. Condition **cond7** includes **GroupA** as a destination group. Flows coming into the switch destined for any of the specified IP addresses in the group will match rule **FilterL32**. **FilterL32** is configured with an action (**Ok**) to allow the traffic on the switch.

Multicast Filtering ACLs

Multicast filtering may be set up to filter clients requesting group membership via the Internet Group Management Protocol (IGMP). IGMP is used to track multicast group membership. The IP Multicast Switching (IPMS) function in the switch optimizes the delivery of IP multicast traffic by sending packets only to those stations that request it. Potential multicast group members may be filtered out so that IPMS does not send multicast packets to those stations.

For more information about IPMS, see Chapter 26, “Configuring IP Multicast Switching.”

Multicast traffic has its own global disposition. By default, the global disposition is **accept**. To change the default, use the **qos default multicast disposition** command.

For multicast filtering, the switch classifies traffic based on the multicast IP address or multicast network group and any destination parameters. Note that the destination parameters are used for the client from which the switch will receive the IGMP request.

The **multicast ip** or **multicast network group** keyword is required in the condition configured for a multicast ACL.

The following keywords may be used in the condition to indicate the client parameters:

Multicast ACL Keywords
destination ip
destination vlan
destination port
destination port group
destination mac
destination mac group
destination interface type

If a destination group is specified, the corresponding single value keyword cannot be combined in the same condition. For example, if a destination port is specified, a destination port group cannot be specified in the same condition.

To filter multicast clients, specify the multicast IP address, which is the address of the multicast group or stream, and specify the client IP address, VLAN, MAC address, or slot/port. For example:

```
-> qos default multicast disposition deny
-> policy condition Mclient1 multicast ip 224.0.1.2 destination vlan 5
-> policy action ok disposition accept
-> policy rule Mrule condition Mclient1 action ok
```

In this example, any traffic coming in on VLAN 5 requesting membership to the 224.0.1.2 multicast group will be allowed.

Using ACL Security Features

The following additional ACL features are available for improving network security and preventing malicious activity on the network:

- **UserPorts**—A port group that identifies its members as user ports to prevent spoofed IP traffic. When a port is configured as a member of this group, packets received on the port are dropped if they contain a source IP network address that does not match the IP subnet for the port. See "Configuring a UserPorts Group" on page 25-18.
- **DropServices**—A service group that improves the performance of ACLs that are intended to deny packets destined for specific TCP/UDP ports. Using the DropServices group for this function minimizes processing overhead, which otherwise could lead to a DoS condition for other applications trying to use the switch. See "Configuring a DropServices Group ACL" on page 25-18.
- **ICMP drop rules**—Allows condition combinations in policies that will prevent user pings, thus reducing DoS exposure from pings. See "Configuring ICMP Drop Rules" on page 25-20.
- **BPDUShutdownPorts**—A port group that identifies its members as ports that should not recevie BPDUs. If a BPDU is received on one of these ports, the port is administratively disabled. See "Configuring a BPDUShutdownPorts Group" on page 25-20.
- **Early ARP discard**—ARP packets destined for other hosts are discarded to reduce processing overhead and exposure to ARP DoS attacks. No configuration is required to use this feature, it is always available and active on the switch. Note that ARPs intended for use by a local subnet, AVLAN, VRRP, and Local Proxy ARP are *not* discarded.

Configuring a UserPorts Group

To prevent IP address spoofing, add ports to a port group called UserPorts. For example, the following **policy port group** command adds ports 1/1-24, 2/1-24, 3/1, and 4/1 to the **UserPorts** group:

```
-> policy port group UserPorts 1/1-24 2/1-24 3/1 4/1
-> qos apply
```

Note that the UserPorts group only applies to routed traffic and it is *not* necessary to include the UserPorts group in a condition and/or rule for the group to take effect. Once ports are designated as members of this group, IP spoofed traffic is blocked while normal traffic is still allowed on the port.

Configuring a DropServices Group ACL

To drop packets destined to specific TCP and UDP ports using minimal switch resources, create an ACL using the DropServices group. This group can be used with two types of conditions: one based on physical source ports and one based on source VLANs. If a source VLAN condition is used, then packets received on ports associated with that VLAN are not blocked if they are destined for any of the services in the DropServices group.

Note that if a source port group condition is used, only a drop action is allowed. If a source VLAN condition is used, only an accept action is allowed.

Use the following steps to configure a DropServices ACL that includes a condition for source ports and a condition for an exception VLAN:

1 Create destination port services for the TCP/UDP traffic that you want dropped using the **policy service** command. For example, the following commands create port services for TCP ports 135 and 445 and UDP ports 137, 138, and 445:

```
-> policy service tcp135 destination tcp port 135
-> policy service tcp445 destination tcp port 445
-> policy service udp137 destination udp port 137
-> policy service udp138 destination udp port 138
-> policy service udp445 destination udp port 445
```

2 Add the services created in Step 1 to a service group called **DropServices** using the **policy service group** command. For example:

```
-> policy service group DropServices tcp135 tcp445 udp137 udp138 udp445
```

Note that the DropServices group must be specified using the exact capitalization as shown in the above example.

3 Create a condition with the DropServices group defined in Step 2 and a source port group using the **policy port group** and **policy condition** commands. For example:

```
-> policy port group badGuyPorts 5/1 6/1
-> policy condition badGuys source port group badGuyPorts service group DropServices
```

4 Create a condition with the DropServices group defined in Step 2 and a source VLAN. For example:

```
-> policy condition goodGuys source vlan 20 service group DropServices
```

5 Create a drop action and an accept action using the **policy action disposition** command. For example:

```
-> policy action DROP disposition drop
-> policy action ACCEPT disposition accept
```

6 Create a deny rule for the source ports and an accept rule for the source VLAN using the **policy rule** command. For example:

```
-> policy rule r1 condition badGuys action DROP
-> policy rule r2 condition goodGuys action ACCEPT
```

7 Apply the QoS configuration using the **qos apply** command.

```
-> qos apply
```

The resulting ACL will drop traffic received on the specified source ports that is destined for service ports identified in the DropServices group, while at the same time allowing traffic originating on VLAN 20 to use these same services. The following shows what this ACL looks like in its entirety:

```
-> policy service tcp135 destination tcp port 135
-> policy service tcp445 destination tcp port 445
-> policy service udp137 destination udp port 137
-> policy service udp138 destination udp port 138
-> policy service udp445 destination udp port 445
-> policy service group DropServices tcp135 tcp445 udp137 udp138 udp445
-> policy port group badGuyPorts 5/1 6/1
-> policy condition badGuys source port group badGuyPorts service group DropServices
-> policy action DROP disposition drop
-> policy action ACCEPT disposition accept
-> policy rule r1 condition badGuys action DROP
-> policy rule r2 condition goodGuys action ACCEPT
```

Configuring ICMP Drop Rules

Combining a Layer 2 condition for source VLAN with a Layer 3 condition for IP protocol is supported. Use these two conditions together in a policy to block ICMP echo request and reply packets without impacting switch performance.

The following example defines an ACL policy that blocks ICMP echo request and reply packets on source VLAN 10:

```
-> policy condition ping10 source vlan 10 ip protocol 1
-> policy action drop disposition drop
-> policy rule noping10 condition ping10 action drop
```

Note that the above policy only blocks ICMP echo traffic, all other ICMP traffic is still allowed.

Configuring a BPDUShutdownPorts Group

To block BPDUs on certain ports, add the desired ports to a port group called BPDUShutdownPorts. For example, the following **policy port group** command adds ports 3/1-24 and 4/1-24 to the **BPDUShutdownPorts** group:

```
-> policy port group BPDUShutdownPorts 3/1-24 4/1-24
-> qos apply
```

Note that it is *not* necessary to include the BPDUShutdownPorts group in a condition and/or rule for the group to take affect. In addition, this group must be specified using the exact capitalization shown in the above example.

Once ports are designated as members of the BPDUShutdownPorts group, BPDUs are blocked by administratively shutting down a port when the port receives a BPDU. To restore a disabled port to enabled status, disconnect and reconnect the cable or use the **interfaces admin** command to administratively enable the port.

Verifying the ACL Configuration

To display information about ACLs, use the same **show** commands that are used for displaying any QoS policies. These commands include:

show policy condition Displays information about all pending and applied policy conditions or a particular policy condition configured on the switch. Use the **applied** keyword to display information about applied conditions only.

show policy action Displays information about all pending and applied policy actions or a particular policy action configured on the switch. Use the **applied** keyword to display information about applied actions only.

show policy rule Displays information about all pending and applied policy rules or a particular policy rule.

show active policy rule Displays the pending and applied policy rules that are active (enabled) on the switch.

When a **show** command is used to display output for all pending and applied policy configuration, the following characters may appear in the display:

character	definition
+	Indicates that the policy rule has been modified or has been created since the last **qos apply**.
-	Indicates the policy object is pending deletion.
#	Indicates that the policy object differs between the pending/applied objects.

The following example shows all policy rules configured on the switch:

```
-> show policy rule
                 Policy           From Prec  Enab Inact  Refl Log  Save
my_rule                           cli     0  Yes   Yes    No  No   Yes
Cnd/Act:                          cond5 -> action2

+my_rule5                         cli     0  Yes  No     No   No   Yes
Cnd/Act:                          cond2 -> pri2

mac1                              cli     0  Yes  No     No   No   Yes
Cnd/Act:                          dmac1 -> pri2
```

The display indicates that **my_rule** is inactive and is not used to classify traffic on the switch (the Inact field displays **Yes**). The rule **my_rule5** has been configured since the last **qos apply** command was entered, as indicated by the plus (+) sign. The rule will not be used to classify traffic until the next **qos apply**. Only **mac1** is actively being used on the switch to classify traffic.

To display only policy rules that are active (enabled) on the switch, use the **show active policy rule** command. For example:

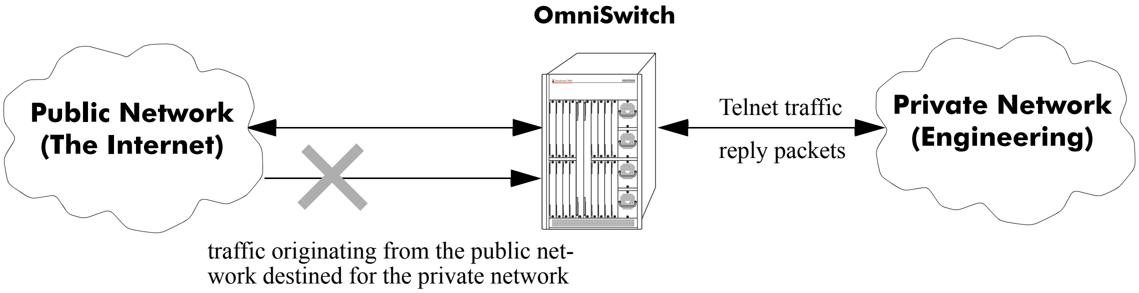
```
-> show active policy rule

                 Policy          From Prec  Enab Inact Refl Log  Save    Matches
+my_rule5                        cli     0  Yes  No    No   No   Yes           0
Cnd/Act:                         cond2 -> pri2

mac1                             cli     0  Yes  No    No   No   Yes           0
Cnd/Act:                         dmac1 -> pri2
```

In this example, the rule **my_rule** does not display because it is inactive. Rules are inactive if they are administratively disabled through the **policy rule** command, or if the rule cannot be enforced by the current hardware. Both **my_rule5** and **mac1** are displayed here because they are active; however, **my_rule5** is a pending rule and will not be used to classify traffic until the **qos apply** command is entered.

See the OmniSwitch *CLI Reference Guide* for more information about the output of these commands.

ACL Application Example

In this application for IP filtering, Telnet traffic from an engineering group is routed through the OmniSwitch 7800. A policy is configured to accept this traffic on the switch. Another policy denies traffic from the outside world to the engineering group; however, reverse flows (reply packets) in response to the Telnet session are allowed through the switch.


OmniSwitch
Public Network (The Internet)
Telnet traffic
reply packets
Private Network (Engineering)
traffic originating from the public network destined for the private network

Two policies must be created in this example.

Policy 1

Set up a policy rule called **outside** to deny Telnet traffic to the private network.

1 Create a policy service (**traffic_in**) for traffic originating from the well-known Telnet port number 23.

```
-> policy service traffic_in source ip port 23 ip protocol 6
```

2 Create a policy condition (**outside_cond**) that references the service.

```
-> policy condition outside_cond service traffic_in
```

3 Create a policy action (**outside_action**) to deny the traffic.

```
-> policy action outside_action disposition deny
```

4 Then combine the condition and the action in a policy rule (**outside**). Note that the rule is not reflexive.

```
-> policy rule outside condition outside_cond action outside_action
```

An example of what these commands look like together on consecutive command lines:

```
-> policy service traffic_in source ip port 23 ip protocol 6
-> policy condition outside_cond service traffic_in
-> policy action outside_action disposition deny
-> policy rule outside condition outside_cond action outside_action
```

Policy 2

Set up a second rule called **telnet_rule** to accept Telnet traffic from the private network out to the public network.

1 Create a policy service for Telnet (**traffic_out**).

```
-> policy service traffic_out protocol 6 destination ip port 23
```

2 Create a policy condition (**telnet_cond**) for the Telnet traffic using the **traffic_out** service configured in step 1.

```
-> policy condition telnet_cond service traffic_out
```

3 Create a policy action (**telnet_action**) to allow the flow.

```
-> policy action telnet_action disposition accept
```

4 Create a rule (**telnet_rule**) that references the policy condition and the policy action. The rule must also use the keyword **reflexive**, so that reply packets that would normally be denied by the **outside** policy rule will be able to get through.

```
-> policy rule telnet_rule reflexive condition telnet_cond action telnet_action
```

An example of what these commands look like together on consecutive command lines:

```
-> policy service traffic_out protocol 6 destination ip port 23
-> policy condition telnet_cond service traffic_out
-> policy action telnet_action disposition accept
-> policy rule telnet_rule reflexive condition telnet_cond action telnet_action
```

This configuration is not applied on the switch until the **qos apply** command is entered. You can test the rule first by using the **policy classify l3** command. See "Testing Conditions" on page 24-47.

26 Configuring IP Multicast Switching

IP Multicast Switching is a one-to-many communication technique employed by emerging applications such as video distribution, news feeds, conferencing, netcasting, and resource discovery (OSPF, RIP2, BOOTP). Unlike unicast, which sends one packet per destination, multicast sends one packet to all devices in any subnetwork that has at least one device requesting the multicast traffic. Multicast switching also requires much less bandwidth than unicast techniques and broadcast techniques since the source hosts only send one data stream to the ports on which destination hosts that request it are attached.

Destination hosts signal their intent to receive a specific multicast stream by sending a request to do so to a nearby switch using Internet Group Management Protocol (IGMP). The switch then learns on which ports multicast group subscribers are attached and can intelligently deliver traffic only to the respective ports. This mechanism is often referred to as *IGMP snooping* (or *IGMP gleaning*). Alcatel's implementation of IGMP snooping is called IP Multicast Switching (IPMS). IPMS allows OmniSwitch switches to efficiently deliver multicast traffic in hardware at wire speed.

In This Chapter

This chapter describes the basic components of IPMS and how to configure them through the Command Line Interface (CLI). CLI commands are used in the configuration examples; for more details about the syntax of commands, see the *OmniSwitch CLI Reference Guide*.

Configuration procedures described in this chapter include:

- Enabling and disabling IPMS on page 26-8.
- Enabling and disabling IPMS hardware routing on page 26-9.
- Configuring and removing a static neighbor on page 26-10.
- Configuring and removing a static querier on page 26-11.
- Modifying IPMS parameters beginning on page 26-13.

Note. You can also configure and monitor IPMS with WebView, Alcatel's embedded web-based device management application. WebView is an interactive and easy-to-use GUI that can be launched from OmniVista or a web browser. Please refer to WebView's online documentation for more information on configuring and monitoring IPMS with WebView.

IPMS Specifications

The table below lists specifications for Alcatel's IPMS software.

RFCs Supported	RFC 2236 — Internet Group Management Protocol, Version 2 RFC 2933 — Internet Group Management Protocol MIB RFC 3376 — Internet Group Management Protocol, Version 3
Leave Timeout	0 to 4294967295 seconds
Query Interval	0 to 4294967295 seconds
Membership Timeout	0 to 4294967295 seconds
Neighbor Timeout	0 to 4294967295 seconds
Querier Timeout	0 to 4294967295 seconds
Flow Timeout	0 to 65535 seconds
Querier Aging and Election Timeout	0 to 4294967295 seconds
Multicast Priority	urgent, high, medium, or low
Maximum Ingress Bandwidth	1 to 1000 megabits

IPMS Default Values

The table below lists default values for Alcatel's IPMS software.

Parameter Description	Command	Default Value/Comments
Administrative Status	**ip multicast switching**	disabled
IGMP Proxy Version	**ip multicast igmp-proxy-version**	Version 2
Hardware Routing	**ip multicast hardware-routing**	disabled
Leave Timeout	**ip multicast leave-timeout**	1 second
Query Interval	**ip multicast query-interval**	125 seconds
Membership Timeout	**ip multicast membership-timeout**	260 seconds
Neighbor Timeout	**ip multicast neighbor-timeout**	90 seconds
Querier Timeout	**ip multicast querier-timeout**	260 seconds
Flow Timeout	**ip multicast flow-timeout**	120 seconds in OmniSwitch 7700/7800/8800 Switches 64800 seconds in OmniSwitch 6600 series
Querier Aging and Election Timeout	**ip multicast other-querier-timeout**	255
Multicast Priority	**ip multicast priority**	low
Maximum Ingress Bandwidth	**ip multicast max-ingress-bandwidth**	10 megabits

IPMS Overview

A multicast group is defined by a multicast group address, which is a Class D IP address in the range 224.0.0.0 to 239.255.255.255. (Addresses in the range 239.0.0.0 to 239.255.255.255 are reserved for boundaries.) The multicast group address is indicated in the destination address field of the IP header. (See "Reserved Multicast Addresses" on page 26-5 for more information.)

IPMS tracks the source VLAN on which the Internet Group Management Protocol (IGMP) requests are received. The network interfaces verify that a multicast packet is received by the switch on the source (or expected) port. The packet is then switched or routed (if the IP multicast routing software has been installed) to all ports in the list.

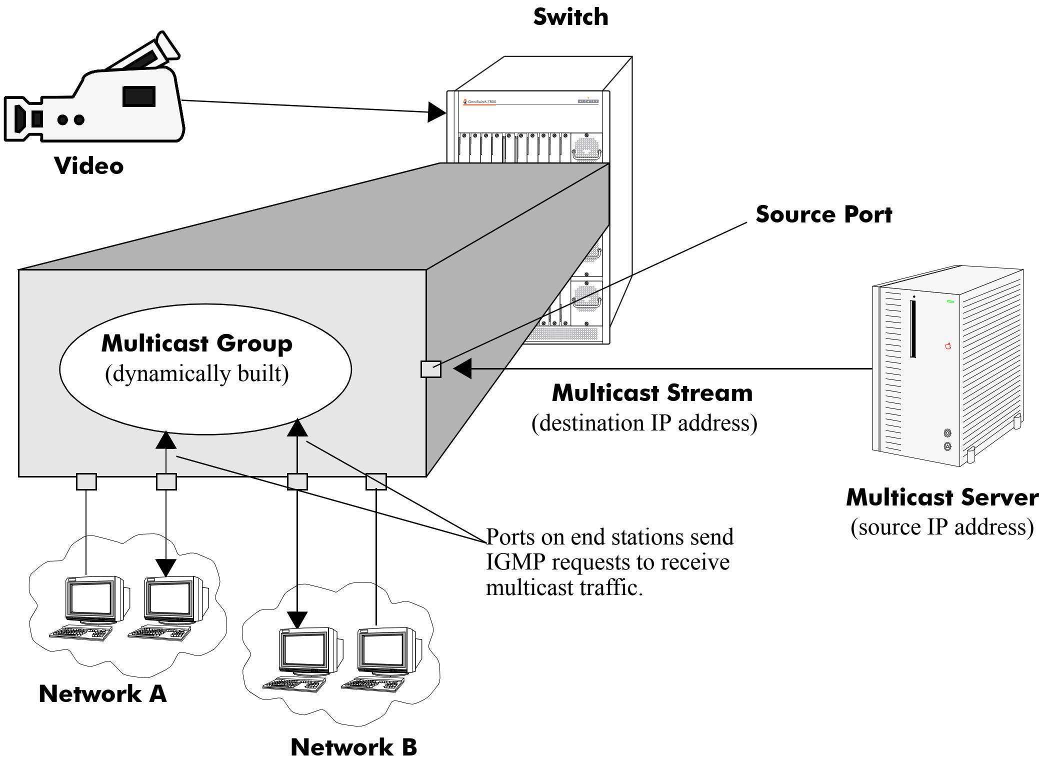
The switching/routing decision is determined dynamically by the IPMS software in the switch. If both ports are in the same VLAN, the packet is switched. If the ports are in different VLANs, the packet is routed if the multicast routing software (see "IP Multicast Routing" on page 26-6) is installed and running on the switch and a valid route is available.

Routing between VLANs on the switch will not be possible if the multicast routing protocols have been disabled. (This may be desirable if you do not want to configure IP addresses for many router ports on the switch.) In this scenario, if both the source and destination ports are in the same VLAN, the packet is switched.

Note. Jumbo multicast packets are not supported. The maximum MTU size supported by Alcatel's IPMS software is 1500.

IPMS Example

The figure on the following page shows an IPMS network where video content can be provided to clients that request it. A server is attached to the switch that provides the source (i.e, multicast) IP addresses. Clients from two different attached networks send IGMP reports to the switch to receive the video content.



Example of an IPMS Network

Reserved Multicast Addresses

The Internet Assigned Numbers Authority (IANA) created the range for multicast addresses, which is 224.0.0.0 to 239.255.255.255. However, as the table below shows, certain addresses are reserved and cannot be used.

Address or Address Range	Description
224.0.0.0 through 224.0.0.255	Routing protocols (e.g., OSPF, RIP2)
224.0.1.0 through 224.0.1.255	Internetwork Control Block (e.g., RSVP, DHCP, commercial servers)
224.0.2.0 through 224.0.255.0	AD-HOC Block (e.g., commercial servers)
224.1.0.0 through 224.1.255.255	ST Multicast Groups
224.2.0.0 through 224.2.255.255	SDP/SAP Block
224.252.000.000 through 224.255.255.255	DIS Transient Groups
225.000.000.000 through 231.255.255.255	Reserved
232.000.000.000 through 232.255.255.255	Source Specific Multicast
233.000.000.000 through 233.255.255.255	GLOP Block
234.000.000.000 through 238.255.255.255	Reserved
239.000.000.000 through 239.255.255.255	Administratively Scoped

IP Multicast Routing

IP multicast routing can be used for IP Multicast Switching and Routing (IPMSR). IP multicast routing is a way of controlling multicast traffic across networks. The multicast router discovers which networks want to receive multicast traffic by sending out Internet Group Management Protocol (IGMP) queries and receiving IGMP reports from attached networks. The IGMP reports signal that users want to join a multicast group.

If there is more than one multicast router in the network, the router with the lowest IP address is elected the querier router, which is responsible for querying the subnetwork for group members.

The IP multicast routing package offers the following two separate protocols:

- Protocol Independent Multicast — Sparse Mode (PIM-SM), which is described in "PIM-SM" on page 26-6.
- Distance Vector Multicast Routing Protocol (DVMRP), which is described in "DVMRP" on page 26-6.

The multicast routing protocols build and maintain a multicast routing database. The multicast routing protocols forward multicast traffic to *networks* that have requested group membership to a specific multicast group. IPMS uses decisions made by the routing protocols and forwards multicast traffic to *ports* that request group membership. See the *OmniSwitch 7700/7800/8800 Advanced Routing Configuration Guide* for more information on IP multicast routing protocols.

PIM-SM

Protocol-Independent Multicast (PIM) is an IP multicast routing protocol that uses routing information provided by unicast routing protocols such as RIP and OSPF. Sparse mode PIM (PIM-SM) contrasts with flood-and-prune dense mode multicast protocols, such as DVMRP and PIM Dense Mode (PIM-DM), in that multicast forwarding in PIM-SM is initiated only via specific requests. Downstream routers must explicitly join PIM-SM distribution trees in order to receive multicast streams on behalf of directly-connected receivers or other downstream PIM-SM routers. This paradigm of receiver-initiated forwarding makes PIM-SM ideal for network environments where receiver groups are thinly populated and bandwidth conservation is a concern, such as in Wide Area Networks (WANs).

DVMRP

Distance Vector Multicast Routing Protocol (DVMRP) is a distributed multicast routing protocol that dynamically generates per-source delivery trees based upon routing exchanges. When a multicast source begins to transmit, the multicast data is flooded down the delivery tree to all points in the network. DVMRP then *prunes* (i.e., removes branches from) the delivery tree where the traffic is unwanted. This is in contrast to PIM-SM, which uses receiver-initiated (i.e., forward path) multicasting.

IPMS and Link Aggregation

When configuring IPMS and link aggregation on the same switch the following conditions should be kept in mind:

- When a port moves to a link aggregation group all IPMS configurations on the port will be lost.
- When the last port in a link aggregation group moves out of the group all IPMS configuration on the link aggregation group will be lost.

IGMP Version 3

IGMP is used by IPv4 systems (hosts and routers) to report their IP multicast group memberships to any neighboring multicast routers. IGMP version 2 (IGMPv2) handles forwarding by IP multicast destination address only. IGMP version 3 (IGMPv3) handles forwarding by source IP address and IP multicast destination address. OmniSwitch 7700/7800/8800 switches support IGMPv2 and IGMPv3.

Note. See "Configuring the IGMP Proxy Version" on page 26-9 for information on configuring the IGMP version.

In IGMPv2, each membership report contains only one multicast group. In IGMPv3, membership reports contain many multicast groups up to the Maximum Transmission Unit (MTU) size of the interface. IGMPv3 uses source filtering and reports multicast memberships to neighboring routers by sending membership reports. IGMPv3 also supports Source Specific Multicast (SSM) by allowing hosts to report interest in receiving packets only from specific source addresses or from all but specific source addresses.

Note. It should be noted that In the current release SSM packet forwarding is not done between ports in the same VLAN. However, SSM forwarding between different VLANs (routing) is supported. In addition, the current implementation of IGMPv3 and SSM only forwards packets to a list of included sources for a given multicast destination. Exclude list forwarding is not supported as it is not a requirement for SSM, and specifically Protocol Independent Multicast–Source Specific Multicast (PIM-SSM). See the *OmniSwitch 7700/7800/8800 Advanced Routing Configuration Guide* for more information on SSM.

IPMS and Switch Fabric Back Pressure

With *back pressure*, a Network Interface (NI) receiving too much multicast traffic will limit transmission of multicast traffic by NIs sending traffic. Without back pressure an NI receiving multicast traffic will silently drop traffic. In the current release, the OmniSwitch's switching fabric does not have any back pressure turned on for multicast traffic. To turn on switch fabric back pressure use the **debug interfaces set backpressure** command. (Use the **debug interfaces backpressure** command to display if this feature has been enabled or not.)

Leaving switch back pressure disabled can cause problems if 802.1Q is used with IPMS because the OmniSwitch 7700/7800/8800 does not support multicasting out a port more than once in hardware. In an 802.1Q configuration, a multicast packet can go out a port multiple times (once for each VLAN). Consequently, IPMS must handle all forwarding of 802.1Q traffic in software. Alcatel therefore recommends enabling switch back pressure with the **debug interfaces set backpressure** command if you plan to use 802.1Q and IPMS together.

Configuring IPMS on a Switch

This section describes how to use Command Line Interface (CLI) commands to enable and disable IP Multicast Switching (IPMS) switch wide (see "Enabling and Disabling IPMS on a Switch" on page 26-8), enable or disable hardware routing switch wide (see "Enabling and Disabling Hardware Routing" on page 26-9), configure a port as a static neighbor (see "Configuring and Removing a Static Neighbor" on page 26-10), and configure a port as a static querier (see "Configuring and Removing a Static Querier" on page 26-11).

Note. Enabling the maximum multicast flood rate with the **interfaces flood multicast** command will limit IPMS and non-IPMS multicast traffic. Either do not use this command with IPMS or set flood rates (set with the **interfaces flood rate** command) high enough to accommodate both flood and IPMS traffic.

Note. IPMS packets should not be routed between IP interfaces on the same VLAN. Doing so will result in the software receiving two (or more) copies of the same multicast frame.

In addition, a tutorial is provided in "IPMS Application Example" on page 26-18 that shows how to use CLI commands to configure a sample network.

Note. See the "IP Multicast Switching Commands" chapter in the *CLI Reference Guide* for complete documentation of IPMS CLI commands.

Enabling and Disabling IPMS on a Switch

IPMS is disabled by default on a switch. The following subsections describe how to enable and disable IPMS with the **ip multicast switching** command.

Note. You must enable or disable IPMS on an entire switch. You cannot enable IPMS on a per port or per slot basis.

Enabling IPMS

To enable IPMS on a switch you use the **ip multicast switching** command as shown below:

```
-> ip multicast switching
```

Disabling IPMS

To disable IPMS you use the **no** form of the **ip multicast switching** command as shown below:

```
-> no ip multicast switching
```

Configuring the IGMP Proxy Version

By default, the version of Internet Group Management Protocol (IGMP) membership is Version 2. The following subsections describe how to configure the proxy version as Version 2 or Version 3 with the **ip multicast igmp-proxy-version** command.

Configuring IGMP Proxy Version 3

To change the IGMP proxy version to Version 3 (IGMPv3) use the **ip multicast igmp-proxy-version** command by entering:

```
-> ip multicast igmp-proxy-version v3
```

Configuring IGMP Proxy Version 2

To restore the IGMP proxy version to Version 2 (IGMPv2) use the **ip multicast igmp-proxy-version** command by either use the **no** form of the command by entering:

```
-> ip multicast no igmp-proxy-version
```

Or, as an alternative enter:

```
-> ip multicast igmp-proxy-version v2
```

To restore the IGMP proxy version to Version 2.

Enabling and Disabling Hardware Routing

By default, IP multicast hardware routing is disabled switch wide. The following subsections describe how to enable and disable hardware routing on a switch with the **ip multicast hardware-routing** command.

Note. You must enable or disable hardware routing on an entire switch. You cannot enable hardware routing on a per port or per slot basis.

Enabling Hardware Routing

To enable hardware routing you use the **ip multicast hardware-routing** command as shown below:

```
-> ip multicast hardware-routing
```

The number of multicast flows supported per Network Interface (NI) is based on the number of queues available in the NI. By default, approximately 400 unique flows are supported (4 queues are allocated per port). This can be increased to approximately 1100 flows per NI by changing the configuration to 2 queues per port. See Chapter 24, “Configuring QoS,” for more information on configuring queues.

In addition, the following limitations should be noted when enabling hardware routing:

- IPMS will route in software to a 802.1Q tagged port, even if hardware routing is enabled.
- Multicast packets cannot be fragmented.
- IP Ethernet-II packets cannot be translated to IP Subnetwork Access Protocol (SNAP) and vice versa.

In all configurations absent of these above limitations, multicast routing can be done in hardware at wire speed.

Disabling Hardware Routing

To disable hardware routing you use the **no** form of the **ip multicast hardware-routing** command as shown below:

```
-> ip multicast no hardware-routing
```

Configuring and Removing a Static Neighbor

IPMS static neighbor ports receive all multicast streams on the designated VLAN and also receive IGMP reports for the VLAN. The following subsections describe how to configure and remove a static neighbor port with the **ip multicast static-neighbor** command.

Configuring a Static Neighbor

You can configure a port as an IPMS static neighbor port by entering **ip multicast static-neighbor** followed by the VLAN number (which must be between 0 and 4095), a space, the slot number of the port, a slash (/), and the port number.

For example, to configure port 4 in slot 10 with designated VLAN 2 as a static neighbor you would enter:

```
-> ip multicast static-neighbor 2 4/10
```

You can also configure a link aggregation group as an IPMS static neighbor port by entering **ip multicast static-neighbor** followed by the VLAN number (which must be between 0 and 4095), **linkagg**, a space, and the aggregation group number.

For example, to configure link aggregation group 7 with designated VLAN 2 as a static neighbor you would enter:

```
-> ip multicast static-neighbor 2 linkagg 7
```

By default, the version of IGMP that will be used is Version 2. As an option, you can also specify the version of IGMP (either version 2 or Version 3) by entering **ip multicast static-neighbor** followed by the VLAN number (which must be between 0 and 4095), a space, either the port (designate the slot number of the port, a slash (/), and the port number) or **linkagg** and the link aggregation group number, and either **v2** or **v3**.

For example, to configure port 4 in slot 10 with designated VLAN 2 as a static neighbor that uses IGMP Version 3 you would enter:

```
-> ip multicast static-neighbor 2 4/10 v3
```

Removing a Static Neighbor

To reset the port so that it is no longer an IPMS static neighbor port you use the **no** form of the **ip multicast static-neighbor** command by entering **ip multicast no static-neighbor** followed by the VLAN number, a space, and either the port (designate the slot number of the port, a slash (/), and the port number) or **linkagg** and the link aggregation group number.

For example, to remove port 4 in slot 10 with designated VLAN 2 as a static neighbor you would enter:

```
-> ip multicast no static-neighbor 2 4/10
```

Configuring and Removing a Static Querier

IPMS static querier ports receive IGMP reports generated on the designated VLAN. Unlike IPMS neighbor ports, they will not receive all multicast streams. The following subsections describe how to configure and remove a static querier with the **ip multicast static-querier** command.

Configuring a Static Querier

You can configure a port as an IPMS static querier port by entering **ip multicast static-querier** followed by the VLAN number (which must be between 0 and 4095), a space, the slot number of the port, a slash (/), and the port number.

For example, to configure port 4 in slot 10 with designated VLAN 2 as a static querier you would enter:

```
-> ip multicast static-querier 2 4/10
```

You can also configure a link aggregation group as an IPMS static querier port by entering **ip multicast static-querier** followed by the VLAN number (which must be between 0 and 4095), **linkagg**, a space, and the link aggregation group number.

For example, to configure link aggregation group 7 with designated VLAN 2 as a static querier you would enter:

```
-> ip multicast static-querier 2 linkagg 7
```

By default, the version of IGMP that will be used is Version 2. As an option, you can also specify the version of IGMP (either version 2 or Version 3) by entering **ip multicast static-querier** followed by the VLAN number (which must be between 0 and 4095), a space, either the port (designate the slot number of the port, a slash (/), and the port number) or **linkagg** and the link aggregation group number, and either **v2** or **v3**.

For example, to configure port 4 in slot 10 with designated VLAN 2 as a static querier that uses IGMP Version 3 you would enter:

```
-> ip multicast static-querier 2 4/10 v3
```

Removing a Static Querier

To reset the port so that it is no longer an IPMS static querier port you use the **no** form of the **ip multicast static-querier** command by entering **ip multicast no static-querier** followed by the VLAN number, a space, the slot number of the port, a slash (/), and either the port (designate the slot number of the port, a slash (/), and the port number) or **linkagg** and the link aggregation group number.

For example, to remove port 4 in slot 10 with designated VLAN 2 as a static querier you would enter:

```
-> ip multicast no static-querier 2 4/10
```

Configuring and Removing a Static Member

The following subsections describe how to configure and remove a static member with the **ip multicast static-member** command.

Configuring a Static Member

You can configure a port as an IPMS static member by entering **ip multicast static-member** followed by the IP address of the static member in dotted decimal notation, a space, the VLAN number (which must be between 0 and 4095), a space, the slot number of the port, a slash (/), and the port number.

For example, to configure a static member with an IP address of 11.0.0.1 on port 10 in slot 3 with designated VLAN 3 you would enter:

```
-> ip multicast static-member 11.0.0.1 3 3/10
```

You can also configure a link aggregation group as an IPMS static member by entering **ip multicast static-member** followed by the IP address of the static member in dotted decimal notation, a space, the VLAN number (which must be between 0 and 4095), a space, **linkagg**, and the link aggregation group number.

For example, to configure a static member with an IP address of 11.0.0.1 on link aggregation group 7 with designated VLAN 3 you would enter:

```
-> ip multicast static-member 11.0.0.1 3 linkagg 7
```

Removing a Static Member

To reset the port so that it is no longer an IPMS static member port you use the **no** form of the **ip multicast static-member** command by entering **ip multicast no static-member** followed by the IP address of the static member, a space, the VLAN number, a space, and either the port (designate the slot number of the port, a slash (/), the port number) or **linkagg** and the link aggregation group number.

For example, to remove a static member with an IP address of 11.0.0.1 on port 10 in slot 3 with designated VLAN 3 you would enter:

```
-> ip multicast no static-neighbor 11.0.0.1 3 3/10
```

Modifying IPMS Parameters

The table in "IPMS Default Values" on page 26-3 lists default values for IPMS parameters. The following sections describe how to use CLI commands to modify these parameters.

Modifying the Leave Timeout

The IPMS leave timeout is the delay in removing a group membership after a leave message has been processed and/or received. The default IPMS leave timeout is 1 second. The following subsections describe how to configure a user-specified leave timeout value and how to restore it with the **ip multicast leave-timeout** command.

Configuring the Leave Timeout

You can modify IPMS leave timeout from 0 to 4294967295 seconds by entering **ip multicast leave-timeout** followed by the new value. For example, to set the leave timeout to 5 seconds you would enter:

```
-> ip multicast leave-timeout 5
```

Restoring the Leave Timeout

To restore the leave timeout to its default (i.e., 1 second) value you use the **no** form of the **ip multicast leave-timeout** command by entering:

```
-> ip multicast no leave-timeout
```

Modifying the Query Interval

The default IPMS query interval (i.e., the time between IGMP queries) is 125 seconds. The following subsections describe how to configure a user-specified query interval value and how to restore it with the **ip multicast query-interval** command.

Configuring the Query Interval

You can modify the query interval from 0 to 4294967295 seconds by entering **ip multicast query-interval** followed by the new value. For example, to set the query interval to 60 seconds you would enter:

```
-> ip multicast query-interval 60
```

Restoring the Query Interval

To restore the query interval to its default (i.e., 125 seconds) value you use the no form of the **ip multicast query-interval** command by entering:

```
-> ip multicast no query-interval
```

Modifying the Membership Timeout

The default IPMS membership timeout (i.e., the time the switch will wait for an IGMP report before it drops a member from a multicast group) is 260 seconds. The following subsections describe how to configure a user-specified membership timeout value and how to restore it with the **ip multicast membership-timeout** command.

Configuring the Membership Timeout

You can modify the IPMS membership timeout from 0 to 4294967295 seconds by entering **ip multicast membership-timeout** followed by the new value. For example, to set the membership timeout value to 100 seconds you would enter:

```
-> ip multicast membership-timeout 100
```

Restoring the Membership Timeout

To restore the membership timeout to its default (i.e., 260 seconds) value you use the **no** form of the **ip multicast membership-timeout** command by entering:

```
-> ip multicast no membership-timeout
```

Modifying the Neighbor Timeout

The default IPMS neighbor timeout (i.e., the time the switch will wait for a neighbor probe from a router before it removes the corresponding entry for the router from the neighbor table) is 90 seconds. The following subsections describe how to configure a user-specified neighbor timeout value and how to restore it with the **ip multicast neighbor-timeout** command.

Configuring the Neighbor Timeout

You can modify the IPMS neighbor timeout from 0 to 4294967295 seconds by entering **ip multicast neighbor-timeout** followed by the new value. For example, to set the neighbor timeout to 360 seconds you would enter:

```
-> ip multicast neighbor-timeout 360
```

Restoring the Neighbor Timeout

To restore the neighbor timeout to its default (i.e., 90 seconds) value you use the **no** form of the **ip multicast neighbor-timeout** command by entering **ip multicast no neighbor-timeout** as shown below:

```
-> ip multicast no neighbor-timeout
```

Modifying the Querier Timeout

The default IPMS querier timeout (i.e., the time the switch will wait for an IGMP query from a device before it removes the corresponding entry for the device from the neighbor table) is 260 seconds. The following subsections describe how to configure a user-specified querier timeout value and how to restore it with the **ip multicast querier-timeout** command.

Configuring the Querier Timeout

You can modify the IPMS querier timeout from 0 to 4294967295 seconds by entering **ip multicast querier-timeout** followed by the new value. For example, to set the querier timeout to 360 seconds you would enter:

```
-> ip multicast querier-timeout 360
```

Restoring the Querier Timeout

To restore the querier timeout to its default (i.e., 260 seconds) value you use the **no** form of the **ip multicast querier-timeout** command by entering:

```
-> ip multicast no querier-timeout
```

Modifying the Flow Timeout

The default multicast flow timeout (i.e., the time in seconds a multicast flow entry is retained by a switch after the last packet in the flow is processed) is 120 seconds in OmniSwitch 7700/7800/8800 switches and 64800 seconds in OmniSwitch 6600 series. The following subsections describe how to configure a user-specified multicast flow timeout value and how to restore it with the **ip multicast flow-timeout** command

Configuring the Flow Timeout

You can modify the multicast flow timeout from 0 to 65535 seconds by entering **ip multicast flow-timeout** followed by the new value. For example, to set the flow timeout to 360 seconds you would enter:

```
-> ip multicast flow-timeout 360
```

Restoring the Flow Timeout

To restore the multicast flow timeout to its default (i.e., 120 seconds in OmniSwitch 7700/7800/8800 switches and 64800 seconds in OmniSwitch 6600 series) value use **ip multicast flow-timeout***,* followed by the value 0.

```
-> ip multicast flow-timeout 0
```

Modifying the Querier Aging and Election Timeout

The default IPMS querier aging and election timeout (i.e., the time for which a currently elected querier is aged and a new multicast querier is elected) is 255 seconds. The following subsections describe how to configure a user-specified querier aging and election timeout value and how to restore it with the **ip multicast other-querier-timeout** command.

Configuring the Querier Aging and Election Timeout

You can modify the IPMS querier aging and election timeout from 0 to 4294967295 seconds by entering **ip multicast other-querier-timeout** followed by the new value. For example, to set the querier aging and election timeout to 120 seconds you would enter:

```
-> ip multicast other-querier-timeout 120
```

Restoring the Querier Aging and Election Timeout

To restore the querier aging and election timeout to its default (i.e., 255 seconds) value you use the **no** form of the **ip multicast other-querier-timeout** command by entering:

```
-> ip multicast no other-querier-timeout
```

Modifying the Multicast Priority

The default IP multicast priority (i.e., the IP multicast priority value on all IPMS data queues used in the switch) is low. The following subsections describe how to configure a user-specified multicast priority value and how to restore it with the **ip multicast priority** command.

Note. Multicast packets always have higher priority than unicast packets. During heavy multicast traffic loads, unicast packet loss may result.

Configuring the Multicast Priority

You can modify the IP multicast priority to low, medium, or urgent by entering **ip multicast priority** followed by **low**, **medium**, **high**, or **urgent**. For example, to set the multicast priority to urgent you would enter:

```
-> ip multicast priority urgent
```

Restoring the Multicast Priority

To restore the IP multicast priority to its default (i.e., low) value you use the **no** form of the **ip multicast priority** command by entering:

```
-> ip multicast no priority
```

Modifying the Maximum Ingress Bandwidth

The default IP multicast maximum ingress bandwidth (i.e., the maximum incoming bandwidth on all IPMS data queues used in the switch) is 10 megabits. The following subsections describe how to configure a user-specified maximum ingress bandwidth value and how to restore it with the **ip multicast max-ingress-bandwidth** command.

Configuring the Maximum Ingress Bandwidth

You can modify the IP multicast maximum ingress bandwidth from 1 to 1000 megabits by entering **ip multicast max-ingress-bandwidth** followed by the new value in megabits. For example, to set the maximum ingress bandwidth to 20 megabits you would enter:

```
-> ip multicast max-ingress-bandwidth 20
```

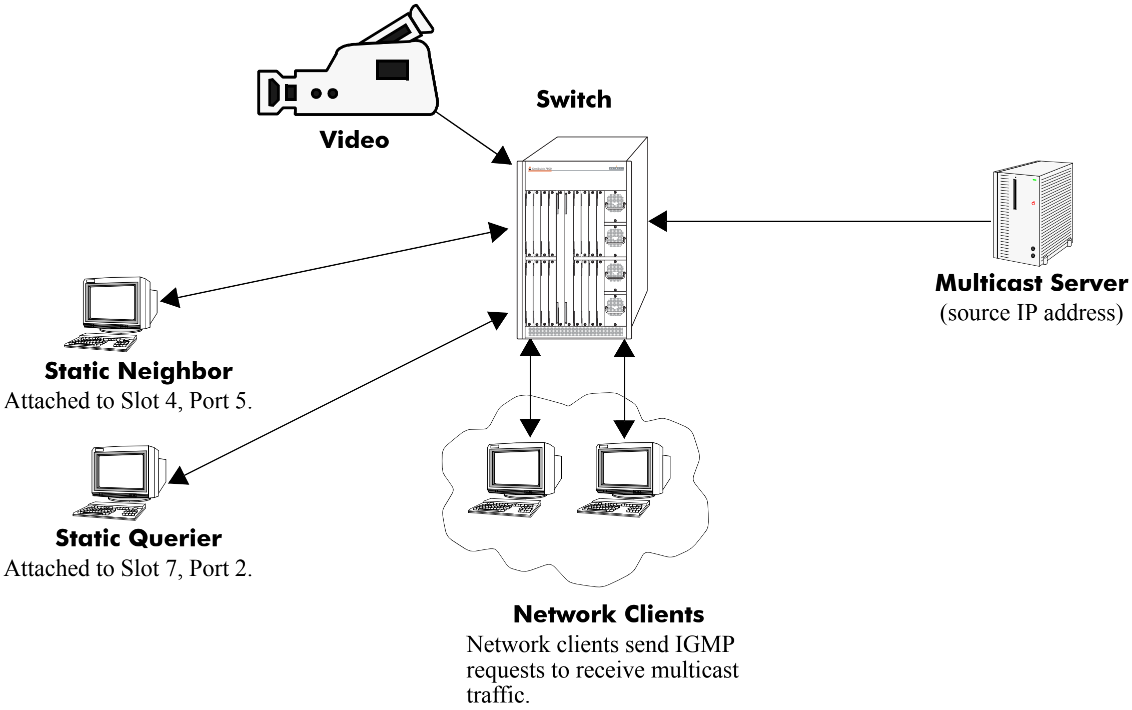
Restoring the Maximum Ingress

To restore the IP multicast maximum ingress bandwidth to its default (i.e., 10 megabits) value you use the **no** form of the **ip multicast max-ingress-bandwidth** command by entering:

```
-> ip multicast no max-ingress-bandwidth
```

IPMS Application Example

The figure below shows a sample network with the switch sending multicast video. A client attached to Slot 4 Port 5 needs to be configured as a static neighbor and another client attached to Slot 7 Port 2 needs to be configured as a static querier.


Video
Switch
Multicast Server
(source IP address)
Static Neighbor
Attached to Slot 4, Port 5.
Static Querier
Attached to Slot 7, Port 2.
Network Clients
Network clients send IGMP requests to receive multicast traffic.

Example of an IMPS Network

The network administrator has determined that due to slow responses from workstations that membership timeout needs to be 6 minutes (i.e., 3600 seconds) and the leave timeout needs to be two minutes (i.e., 120 seconds). In addition, the network administrator has determined that this multicast traffic is not a high priority resource and therefore it needs to be given a low priority within the network.

Follow the steps below to configure this network:

Note. All the steps following Step 1 (which must be executed first) may be entered in any order.

1 Enable IPMS switch wide by entering:

```
-> ip multicast switching
```

2 Configure the client attached to Slot 4 Port 5 as a static neighbor belonging to VLAN 5 by entering:

```
-> ip multicast static-neighbor 5 4/5
```

3 Configure the client attached to Slot 7 Port 2 as a static querier belonging to VLAN 5 by entering:

```
-> ip multicast static-querier 5 7/2
```

4 Modify the membership timeout from its default value of 260 seconds to 3600 seconds by entering:

```
-> ip multicast membership-timeout 3600
```

5 Modify the leave timeout from its default value of 10 seconds to 120 seconds by entering:

```
-> ip multicast leave-timeout 120
```

6 Modify the IPMS priority from high to low by entering:

```
-> ip multicast priority low
```

An example of what these commands look like entered sequentially on the command line:

```
-> ip multicast switching
-> ip multicast static-neighbor 5 4/5
-> ip multicast static-querier 5 7/2
-> ip multicast membership-timeout 3600
-> ip multicast leave-timeout 120
-> ip multicast priority low
```

As an option, you can use the **show ip multicast switching**, **show ip multicast neighbors**, and **show ip multicast queriers** commands to confirm your settings as shown below:

```
-> show ip multicast switching
IPMS Configuration

IPMS State:            Disabled
Hardware Routing:      Enabled
Priority:              low
Max Ingress Bandwidth: 10
Leave Timeout:         120
Flow Timeout:          120
Membership Timeout:    3600
Neighbor Timeout:      90
Querier Timeout:       260
Other Querier Timeout: 255
Query Interval:        125
Default Proxy Version: IGMPv2
Learning Mode:         buffer
```

```
-> show ip multicast neighbors
     Source IP      VLAN Slot/Port Expire  Type    Version
--------------------+----+---------+------+-------+-------
None                   5       4/5  Never  Static  IGMPv2
```

```
-> show ip multicast queriers
     Source IP      VLAN Slot/Port Expire  Type     Version
--------------------+----+---------+------+-------+--------
None                   5       7/2  Never  Static   IGMPv2
```

Displaying IPMS Configurations and Statistics

Alcatel's IP Multicast Switching (IPMS) **show** commands provide tools to monitor IPMS traffic and settings and to troubleshoot problems. These commands are described below:

show ip multicast switching Displays the current IPMS configuration on a switch.

show ip multicast groups Displays all detected multicast groups that have members. If you do not specify an IP address then all multicast groups on the switch will be displayed.

show ip multicast neighbors Displays all neighboring multicast routers.

show ip multicast queriers Displays all multicast queriers.

show ip multicast forwarding Displays the IPMS multicast forwarding table. If you do not specify a multicast group IP address then the forwarding table for all multicast groups will be displayed.

show ip multicast policy-cache Displays the active policies being enforced in the IPMS policy cache.

If you are interested in a quick look at IPMS groups on your switch you could use the **show ip multicast groups** command. For example:

```
-> show ip multicast groups
                                 Source IP         Slot/
Destination IP     Client IP     (IGMPv3 only)  VLAN Port  Expire  Type
---------------+---------------+---------------+----+-----+------+-------
224.0.0.9           11.0.0.1     224.0.0.9        3    3/10  186   Dynamic
225.10.10.10        10.0.0.1                      2     3/9  254   Dynamic
```

Note. See the "IP Multicast Switching Commands" chapter in the *OmniSwitch CLI Reference Guide* for complete documentation on IPMS **show** commands.

27 Configuring Server Load Balancing

Alcatel's Server Load Balancing (SLB) software provides a method to logically manage a group of physical servers sharing the same content (known as a *server farm*) as one large virtual server (known as an *SLB cluster*). SLB clusters are identified and accessed at Layer 3 by the use of Virtual IP (VIP) addresses. OmniSwitch 7700/7800/8800 switches operate at wire speed to process client requests addressed to the VIP of an SLB cluster and send them to the physical servers within the cluster.

Using SLB clusters can provide cost savings (costly hardware upgrades can be delayed or avoided), scalability (as the demands on your server farm grow you can add additional physical servers), reliability (if one physical server goes down the remaining servers can handle the remaining workload), and flexibility (you can tailor workload requirements individually to servers within a cluster).

In This Chapter

This chapter describes the basic components of Server Load Balancing and how to configure them through the Command Line Interface (CLI). CLI commands are used in the configuration examples; for more details about the syntax of commands, see the *OmniSwitch CLI Reference Guide*.

Configuration procedures described in this chapter include:

- Steps to configure physical servers in SLB clusters on page 27-11.
- Procedures to configure SLB on a switch on page 27-24.
- Procedures to configure logical SLB clusters on page 27-25.
- Procedures to configure physical servers on page 27-26.
- Procedures to configure SLB probes on page 27-33.
- Procedures for troubleshooting and maintenance on page 27-31 and page 27-37.

Note. You can also configure and monitor Server Load Balancing with WebView, Alcatel's embedded web-based device management application. WebView is an interactive and easy-to-use GUI that can be launched from OmniVista or a web browser. Please refer to WebView's online documentation for more information on configuring and monitoring Server Load Balancing with WebView.

Server Load Balancing Specifications

The table below lists specifications for Alcatel's SLB software.

Maximum number of clusters	15
Maximum number of physical servers	75
Layer 3 Classification	Source IP address/Destination IP address pairs
Server Health Checking	Ping, link checks
High Availability Support	Hardware-based failover, VRRP, Chassis Management Module (CMM) redundancy
Networking Protocols Supported	Virtual IP (VIP) addresses
Distribution Algorithms Supported	weighted round robin, server failover
"Sticky" Time Range	0 to 86400 seconds
Ping Period Range	0 to 3600 seconds
Ping Timeout range	0 to 1000 times the value of the ping period
Ping Retries	0 to 255
Relative Weight of Physical Servers in an SLB Cluster	0 to 100
Maximum number of probes on a switch	20
Probe timeout range	0 to 3600000 seconds
Probe period range	0 to 3600
Probe port range	0 to 65535
Probe retry range	0 to 255
Probe status range	0 to 4294967295

Server Load Balancing Default Values

The table below lists default values for Alcatel's SLB software.

Parameter Description	Command	Default Value/Comments
Global SLB Administrative Status	**ip slb admin**	disabled
Distribution Algorithm	**ip slb cluster distribution**	weighted round robin
"Sticky" Time (i.e., the maximum delay with no activity a client is kept attached to a physical server)	**ip slb cluster sticky time**	1200 seconds
Ping Period	**ip slb cluster ping period**	60 seconds
Ping Timeout	**ip slb cluster ping timeout**	3000 milliseconds
Ping Retries	**ip slb cluster ping retries**	3
Relative Weight of Physical Servers in an SLB Cluster	**ip slb server ip cluster**	10
Administrative Status of an SLB Cluster	**ip slb cluster admin status**	enabled
Administrative Status of Physical Servers in an SLB Cluster	**ip slb server ip cluster**	enabled
SLB probes configured	**ip slb probe**	None configured
SLB probe timeout	**ip slb probe timeout**	3000 seconds
SLB probe period	**ip slb probe period**	60 seconds
SLB probe port number	**ip slb probe port**	0
SLB probe retries	**ip slb probe retries**	3
SLB probe user name	**ip slb probe username**	None configured
SLB probe password	**ip slb probe password**	None configured
SLB probe URL	**ip slb probe url**	None configured
SLB probe expected status	**ip slb probe status**	200
SLB probe send string	**ip slb probe send**	None configured
SLB probe expect string	**ip slb probe expect**	None configured

Quick Steps for Configuring Server Load Balancing (SLB)

Follow the steps below for a quick tutorial on configuring mandatory parameters for SLB. Additional information on how to configure each command is given in the subsections that follow.

1 Enable SLB globally with the **ip slb admin** command as shown below:

```
-> ip slb admin enable
```

2 Configure the SLB cluster with the **ip slb cluster** command. For example:

```
-> ip slb cluster WorldWideWeb vip 128.241.130.204
```

3 Assign physical servers to the SLB cluster with the **ip slb server ip cluster** command. For example:

```
-> ip slb server ip 128.241.130.127 cluster WorldWideWeb
-> ip slb server ip 128.241.130.109 cluster WorldWideWeb
-> ip slb server ip 128.241.130.115 cluster WorldWideWeb
-> ip slb server ip 128.241.130.135 cluster WorldWideWeb admin status disable
```

As an option, you can verify your SLB settings by entering **show ip slb cluster** followed by the name of the SLB cluster. For example:

```
-> show ip slb cluster WorldWideWeb
Cluster WorldWideWeb
  VIP                              : 128.241.130.204,
  Admin status                     : Enabled,
  Operational status               : In Service,
  Routed flows success ratio (%)   = 100,
  Ping period (seconds)            = 60,
  Ping timeout (milliseconds)      = 3000,
  Ping retries                     = 3,
  Redirect algorithm               : round robin,
  Sticky time (seconds)            = 1200,
  Number of flows                  = 3800,
  Number of servers                = 4
  Server 128.241.130.109
     Admin status = Enabled, Operational Status = In Service,
     Weight = 10, Number of flows = 2000, Availability (%) = 100
  Server 128.241.130.115
     Admin status = Enabled, Operational Status = In Service,
     Weight = 10, Number of flows = 1800, Availability (%) = 98
  Server 128.241.130.127
     Admin status = Enabled, Operational Status = Discovery,
     Weight = 10, Number of flows = 0, Availability (%) = 0
  Server 128.241.130.135
     Admin status = Disabled, Operational Status = Disabled,
     Weight = 10, Number of flows = 0, Availability (%) = 0
```

An example of what these commands look like entered sequentially on the command line:

```
-> ip slb admin enable
-> ip slb cluster WorldWideWeb vip 128.241.130.204
-> ip slb server ip 128.241.130.127 cluster WorldWideWeb
-> ip slb server ip 128.241.130.109 cluster WorldWideWeb
-> ip slb server ip 128.241.130.115 cluster WorldWideWeb
-> ip slb server ip 128.241.130.135 cluster WorldWideWeb admin status disable
```

Server Load Balancing Overview

In the current release, you can add up to 75 physical servers on an OmniSwitch 7700/7800/8800. In addition, you can configure up to 15 Server Load Balancing (SLB) clusters on an OmniSwitch 7700/7800/8800. The following sections describe SLB operational theory (see "Server Load Balancing VIPs" on page 27-6), an SLB example ("Server Load Balancing Example" on page 27-6) SLB distribution algorithms ("Distribution Algorithms" on page 27-7), and server health monitoring (see "Server Health Monitoring" on page 27-10).

Note. Alcatel also offers link aggregation, which combines multiple Ethernet links into one virtual channel. For more information on link aggregation, please refer to Chapter 12, "Configuring Dynamic Link Aggregation," for information on dynamic link aggregation and please refer to Chapter 11, "Configuring Static Link Aggregation," for information on static (OmniChannel) link aggregation.

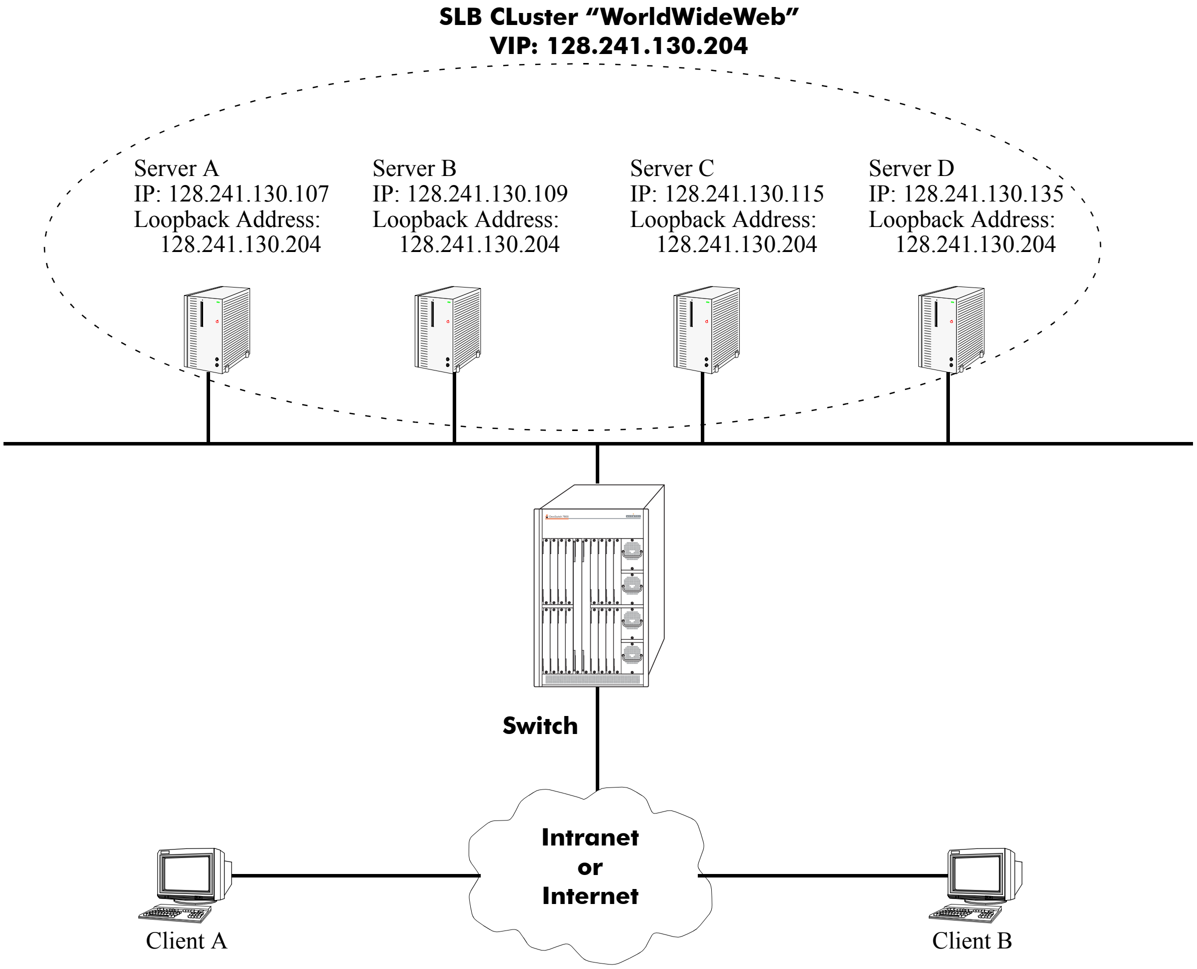
Server Load Balancing VIPs

Server load balancing uses a Virtual IP (VIP) address to treat a group of physical servers, each with a unique IP address, as one large virtual server. The switch will process requests by clients addressed to the VIP of the SLB cluster and send them to the physical servers. This process is totally transparent to the client.

Note. See "Configuring the Server Farm" on page 27-11 for more information on configuring servers.

Server Load Balancing Example

In the figure on the following page, an SLB cluster consisting of four (4) physical servers has been configured with a VIP of 128.241.130.204 and an SLB cluster name of "WorldWideWeb." The switch processes requests sent by clients to the VIP of 128.241.130.204 and sends to the appropriate physical server, depending on configuration and the operational states of the physical servers. The switch then transmits the requested data from the physical server back to the client.



Example of a Server Load Balancing (SLB) Cluster

Distribution Algorithms

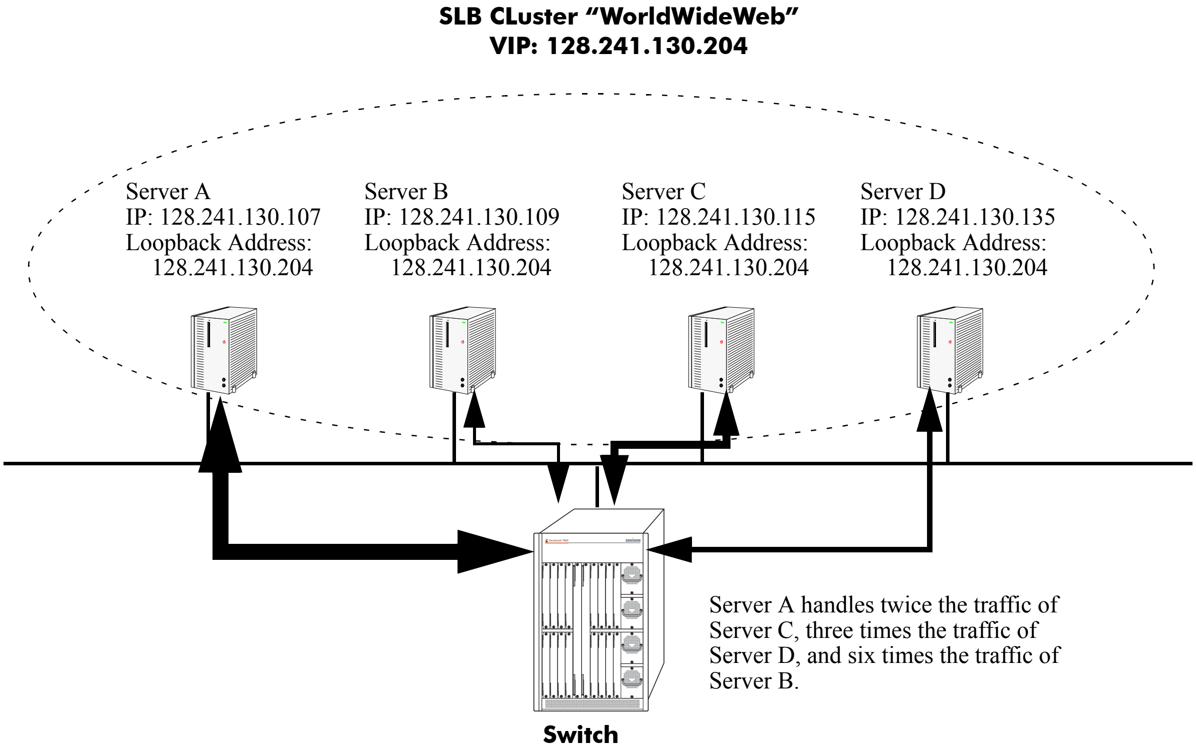
In order to distribute traffic among operating servers, the SLB cluster must have a method of selecting a server among a pool (cluster) of operating servers (i.e., "in service") depending on some criteria. This method is commonly called the SLB cluster's *distribution algorithm.*

Alcatel's SLB software supports two different types of distribution algorithms:

- Weighted round robin—This algorithm is described in "Weighted Round Robin Distribution Algorithm" on page 27-8.
- Server failover—This algorithm is described in "Server Failover Distribution Algorithm" on page 27-9.

Weighted Round Robin Distribution Algorithm

The default distribution algorithm on an Alcatel switch is weighted round robin, where the SLB cluster distributes traffic according to the relative "weight" a server has within an SLB cluster. In the figure below, for example, Server A has been assigned by the network administrator a relative weight of 30, which is the largest weight in the SLB cluster called "WorldWideWeb." Server A will handle twice as much traffic as Server C (which has a weight of 15), three times the traffic of Server D (which has a weight of 10), and six times the traffic of Server B (which has a weight of 5).



Server A handles twice the traffic of Server C, three times the traffic of Server D, and six times the traffic of Server B.

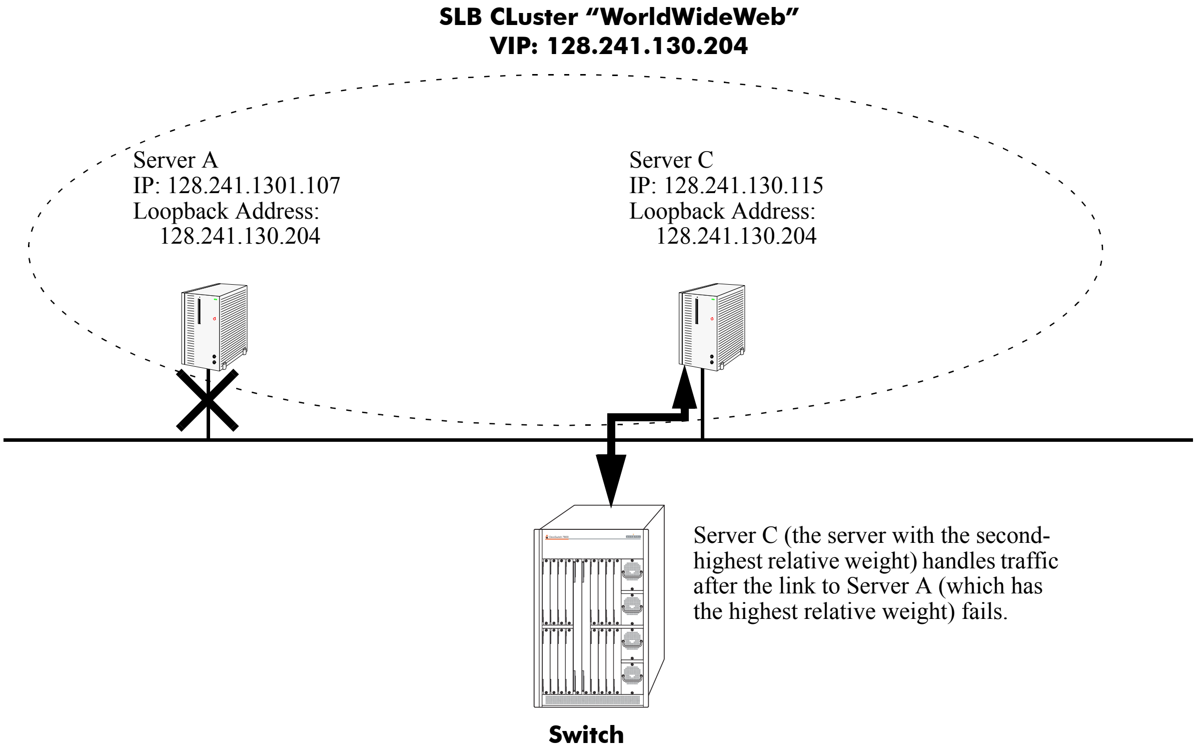
Weighted Round Robin Algorithm

Note. See "Modifying the Relative Weight of a Physical Server" on page 27-30 for information on modifying the relative weights of servers in an SLB cluster.

Server Failover Distribution Algorithm

As an option, you can modify the switch's default distribution algorithm from weighted round robin to server failover. In this algorithm all new client flows are directed towards the highest weighted operational server within the cluster. Other servers within the cluster receive no traffic and serve as backup should the current server be declared as "out of service" (i.e., the server does not respond to ping requests and/or fails link checks). The weights of the servers determine the order in which servers are considered. Identical weights do not guarantee any particular order.

In the figure on the following page, Server A has the highest relative weight and Server C has the second highest. If the link to Server A goes down, Server C will handle all new traffic flows. Server failover therefore provides a quick and easy method for server redundancy.


SLB CLuster "WorldWideWeb"
VIP: 128.241.130.204
Server A
IP: 128.241.1301.107
Loopback Address:
128.241.130.204
Server C
IP: 128.241.130.115
Loopback Address:
128.241.130.204
Server C (the server with the second-highest relative weight) handles traffic after the link to Server A (which has the highest relative weight) fails.
Switch

Server Failover Algorithm

Note. See "Modifying the Distribution Algorithm" on page 27-28 for information on changing the distribution algorithm from weighted round robin to server failover.

Server Health Monitoring

Alcatel's Server Load Balancing (SLB) software on the switch performs checks on the links from the switch to the servers. In addition, the SLB software also sends ICMP echo requests (i.e., ping packets) to the physical servers to determine their availability.

Note. You can use the **show ip slb cluster server** command, which is described in "Displaying Server Load Balancing Status and Statistics" on page 27-37, to display link and ping status of physical servers.

These health checks performed by the switch are used by the SLB software to determine the operational states of servers. The possible operational states are described in the table below:

Operational States

Disabled	The server has been administratively disabled by the user.
No Answer	The server has not responded to ping requests from the switch.
Link Down	There is a bad connection to the server.
Discovery	The switch is pinging a physical server.
In Service	The server can be used for client connections.
Retrying	The switch is making another attempt to bring up the server.

In Release 5.1.6 and later you can configure probes to monitor the health of clusters and servers. See "Configuring SLB Probes" on page 27-33 for more information.

Configuring the Server Farm

To configure a server you must associate the Virtual IP (VIP) address of the Server Load Balancing (SLB) cluster to the loopback interface of the physical server. Otherwise, physical servers will reject packets addressed to the VIP address of the SLB cluster.

Note. A server can be configured with more than one VIP. Therefore, a server can belong to more than one SLB cluster.

This section describes procedures for configuring several commonly-used server operating systems, including Windows NT (see "Configuring a Windows NT Server" on page 27-11), Windows 2000 (see "Configuring a Windows 2000 Server" on page 27-14), Unix and Linux (see "Configuring a Loopback Interface on Unix- and Linux-Based Servers" on page 27-22), and Novell Netware 6 (see "Configuring a Virtual IP Address on a Novell Netware 6 Server" on page 27-23). Please refer to your server's user documentation for operating systems not covered in this chapter.

Note. The following two sections on configuring Windows NT and 2000 servers assume that the Microsoft loopback adapter driver has been installed on your workstation. If you need to install this driver, please refer to "Adding the Microsoft Loopback Adapter Driver" on page 27-16.

Configuring a Windows NT Server

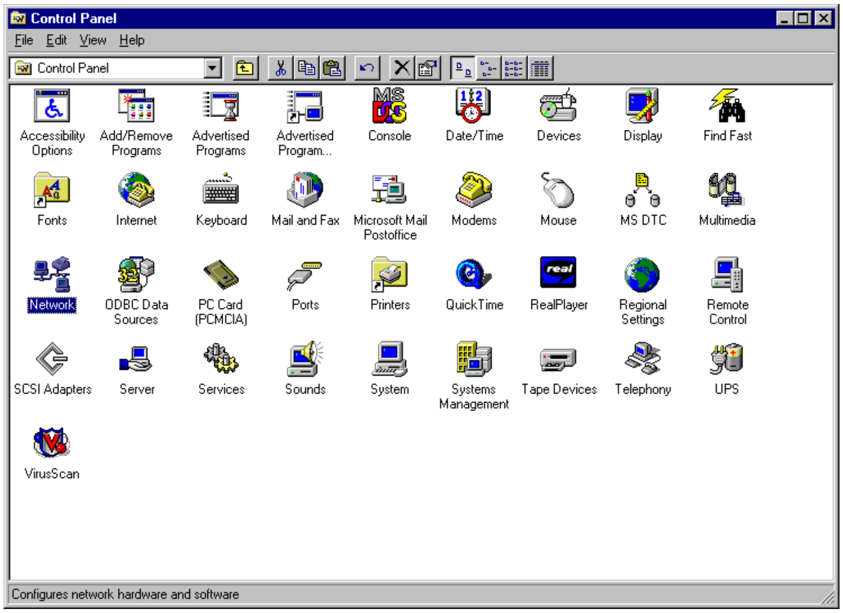
Follow the steps below to associate a loopback interface on a Windows NT server.

Note. This procedure assumes that your Windows NT workstation already has the Microsoft loopback adapter installed. If this driver has not been installed, please perform the steps in "Adding the Loopback Adapter Driver to a Windows NT Server" on page 27-16 before proceeding.

1 Open the **Control Panel** window by clicking the **Start** button and then selecting **Settings**.

2 Double-click the **Network** icon in the **Control Panel** window.

Double click the Network icon.

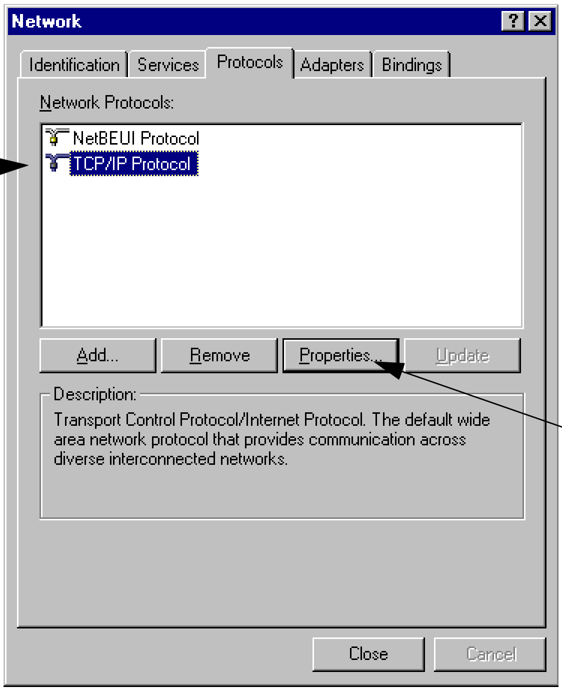


Configures network hardware and software

3 Click the **Protocols** tab in the **Network** window.

4 Select the **TCP/IP Protocol** icon in the **Network Protocols** window.

Select the “TCP/IP Protocol” icon.

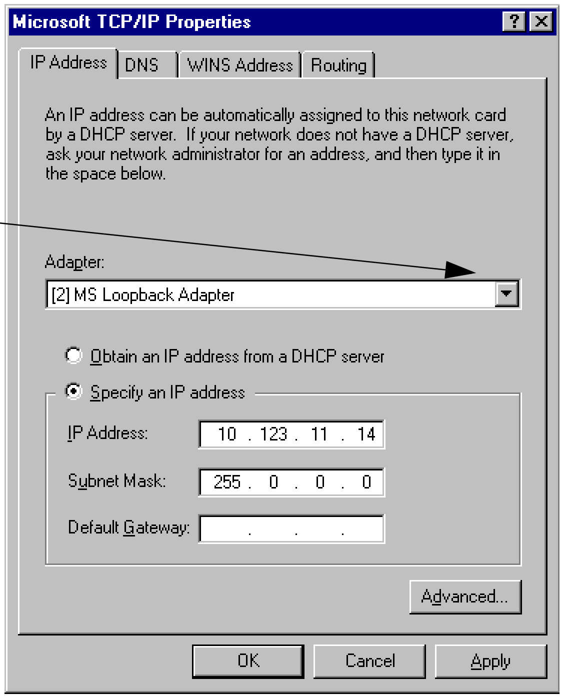


Transport Control Protocol/Internet Protocol. The default wide area network protocol that provides communication across diverse interconnected networks.

Click the “Properties” button when ready.

5 Click the **Properties** button.

6 Select **MS Loopback Adapter** from the **Adapter** pull-down window.



An IP address can be automatically assigned to this network card by a DHCP server. If your network does not have a DHCP server, ask your network administrator for an address, and then type it in the space below.

Adapter: [2] MS Loopback Adapter

○ Obtain an IP address from a DHCP server

● Specify an IP address

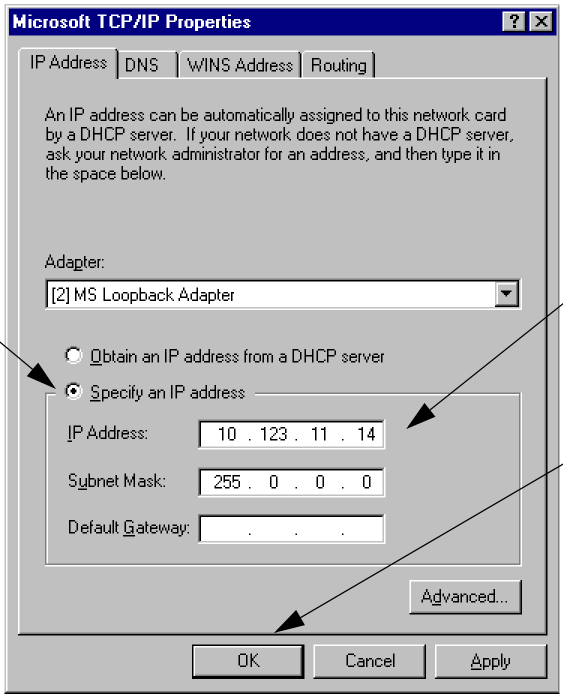
IP Address: 10 . 123 . 11 . 14

Subnet Mask: 255 . 0 . 0 . 0

Default Gateway: . . .

Select the “MS Loopback Adapter” from the “Adapter” pull-down window.

7 Click the **Select an IP address** radio button.



An IP address can be automatically assigned to this network card by a DHCP server. If your network does not have a DHCP server, ask your network administrator for an address, and then type it in the space below.

Adapter: [2] MS Loopback Adapter

○ Obtain an IP address from a DHCP server

● Specify an IP address

IP Address: 10 . 123 . 11 . 14

Subnet Mask: 255 . 0 . 0 . 0

Default Gateway: . . .

Select the “Select an IP address” radio button.

Enter the VIP address here.

Click “OK” when done.

8 Enter the Virtual IP (VIP) address in the **IP Address** window.

Note. Use the same subnet mask as for the physical IP interface.

9 Click the **OK** button.

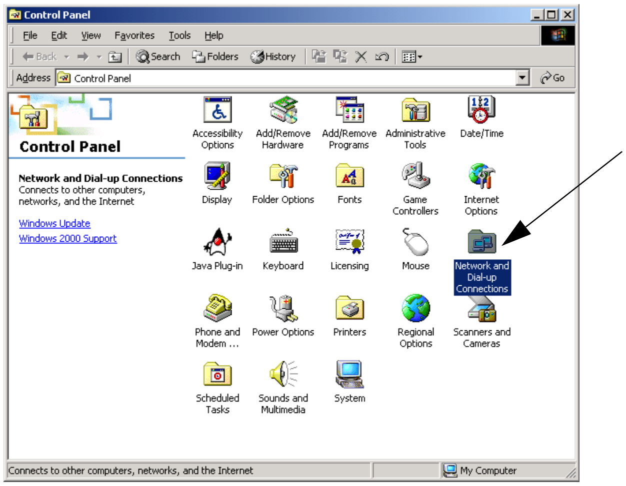
Configuring a Windows 2000 Server

Follow the steps below to associate a loopback interface on a Windows NT server.

Note. This procedure assumes that your Windows 2000 workstation already has the Microsoft loopback adapter installed. If this driver has not been installed, please perform the steps in "Adding the Loopback Adapter Driver to a Windows 2000 Server" on page 27-18 before proceeding.

1 Open the **Control Panel** window by clicking the **Start** button and then selecting **Settings**.

2 Double-click the **Network and Dial-up Connections** icon in the **Control Panel** window.

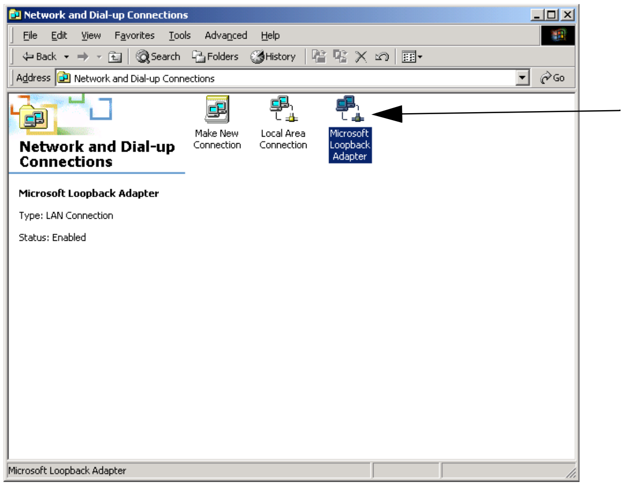


Network and Dial-up Connections
Connects to other computers, networks, and the Internet

Connects to other computers, networks, and the Internet

Double click the "Network and Dial-Up Connections" icon.

3 Right-click the **Microsoft Loopback Adapter** icon in the **Network and Dial-up Connections** window.


Network and Dial-up Connections
File Edit View Favorites Tools Advanced Help
Back Search Folders History
Address Network and Dial-up Connections
Go
Network and Dial-up Connections
Make New Connection
Local Area Connection
Microsoft Loopback Adapter

Microsoft Loopback Adapter

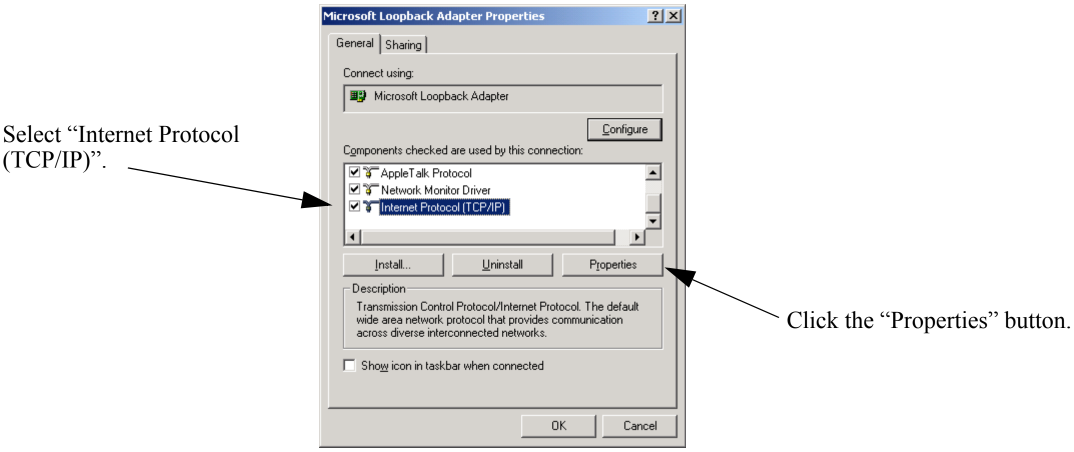
Type: LAN Connection

Status: Enabled

Microsoft Loopback Adapter

Right-click the "Network Loopback Adapter" icon.

4 Select **Internet Protocol (TCP/IP)** in the **Microsoft Loopback Adapter Properties** window.



Microsoft Loopback Adapter Properties

General | Sharing

Connect using:

Microsoft Loopback Adapter

Components checked are used by this connection:

- ☑ AppleTalk Protocol
- ☑ Network Monitor Driver
- ☑ Internet Protocol (TCP/IP)

Description

Transmission Control Protocol/Internet Protocol. The default wide area network protocol that provides communication across diverse interconnected networks.

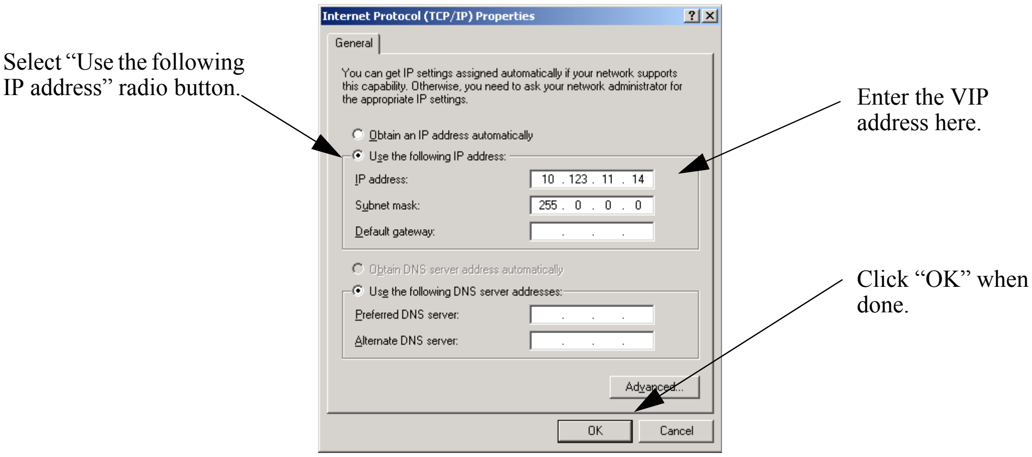
□ Show icon in taskbar when connected

Select "Internet Protocol (TCP/IP)".

Click the "Properties" button.

5 Click the **Properties** button.

6 Click the **Use the following IP address** radio button in the **Internet Properties (TCP/IP) Properties** window.



Internet Protocol (TCP/IP) Properties

General

You can get IP settings assigned automatically if your network supports this capability. Otherwise, you need to ask your network administrator for the appropriate IP settings.

○ Obtain an IP address automatically

● Use the following IP address:

IP address: 10 . 123 . 11 . 14

Subnet mask: 255 . 0 . 0 . 0

Default gateway: . . .

○ Obtain DNS server address automatically

● Use the following DNS server addresses:

Preferred DNS server: . . .

Alternate DNS server: . . .

Select "Use the following IP address" radio button.

Enter the VIP address here.

Click "OK" when done.

7 Enter the Virtual IP (VIP) address in the **IP Address** window.

Note. Use the same subnet mask as for the physical IP interface.

8 Click the **OK** button.

Adding the Microsoft Loopback Adapter Driver

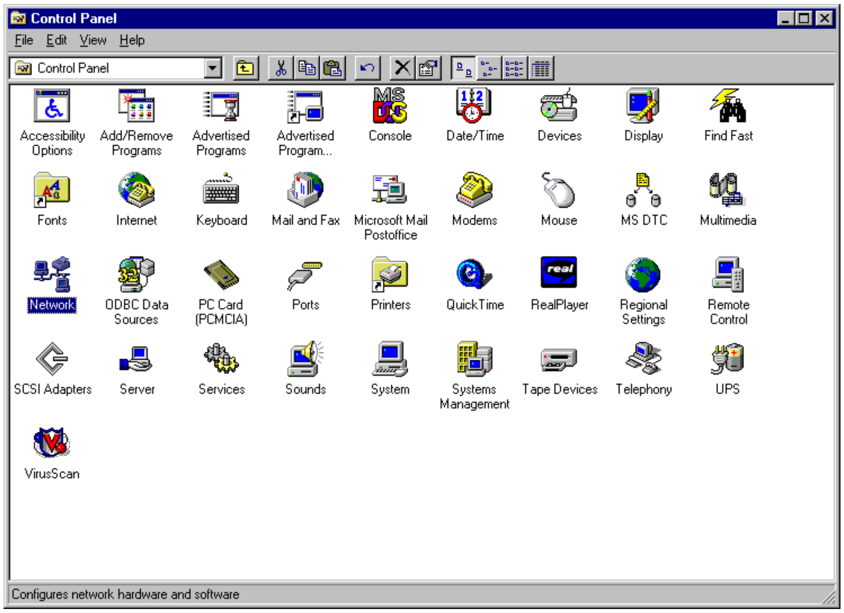
This section describes how to add Microsoft's loopback adapter to Windows NT servers (see "Adding the Loopback Adapter Driver to a Windows NT Server" on page 27-16) and Windows 2000 servers (see "Adding the Loopback Adapter Driver to a Windows 2000 Server" on page 27-18).

Adding the Loopback Adapter Driver to a Windows NT Server

Follow the steps below to add the Microsoft loopback adapter driver to a Windows NT server.

1 Open the **Control Panel** window by clicking the **Start** button and then selecting **Settings**.

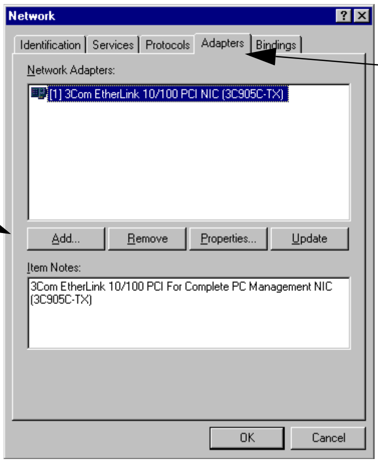
2 Double-click the **Network** icon in the **Control Panel** window.



Configures network hardware and software

Double-click the "Network" icon.

3 Click the **Adapters** tab in the **Network** window.



Network Adapters:

[1] 3Com EtherLink 10/100 PCI NIC (3C905C-TX)

Item Notes:

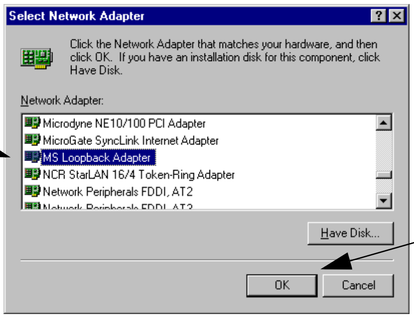
3Com EtherLink 10/100 PCI For Complete PC Management NIC (3C905C-TX)

Click the "Add..." button when ready.

Click the "Adapters" tab.

4 Click the **Add...** button.

5 Select **MS Loopback Adapter** in the **Select Network Adapter** window.



Click the Network Adapter that matches your hardware, and then click OK. If you have an installation disk for this component, click Have Disk.

Network Adapter:

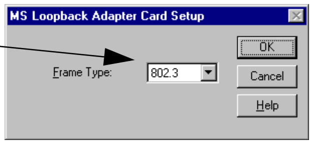
- Microdyne NE10/100 PCI Adapter
- MicroGate SyncLink Internet Adapter
- MS Loopback Adapter
- NCR StarLAN 16/4 Token-Ring Adapter
- Network Peripherals FDDI, AT2

Select "MS Loopback Adapter".

Click "OK" when done.

6 Click the **OK** button.

7 Select the proper frame type in the **Frame Type** pull-down menu.



Frame Type: 802.3

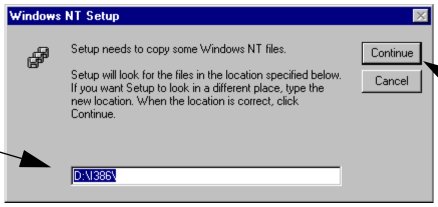
Select "Frame Type".

Click "OK" when done.

8 Click the **OK** button.

9 Load the CD or floppy disc with the Microsoft loopback adapter.

10 If needed, enter the file and path name of the Microsoft loopback adapter in the **Windows NT Setup** window.



Setup needs to copy some Windows NT files.

Setup will look for the files in the location specified below. If you want Setup to look in a different place, type the new location. When the location is correct, click Continue.

D:\I386\

Enter the file and path name here.

Click "Continue" when done.

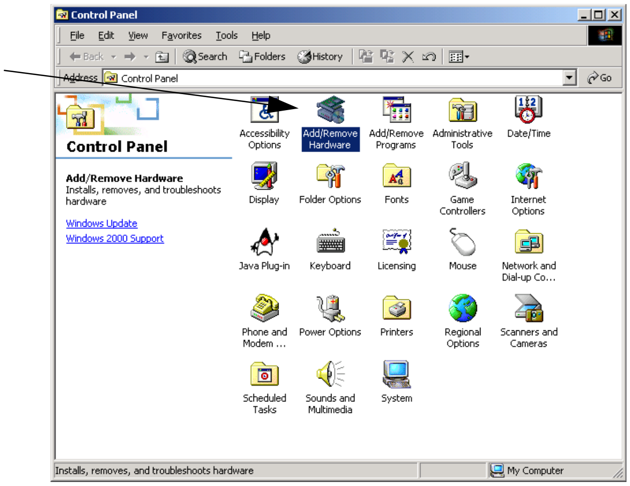
11 Click the **Continue** button. All the necessary files will be copied and installed on your workstation.

Adding the Loopback Adapter Driver to a Windows 2000 Server

Follow the steps below to add the Microsoft loopback adapter driver to a Windows 2000 server.

1 Open the **Control Panel** window by clicking the **Start** button and then selecting **Settings**.

2 Double-click the **Add/Remove Hardware** icon in the **Control Panel** window.



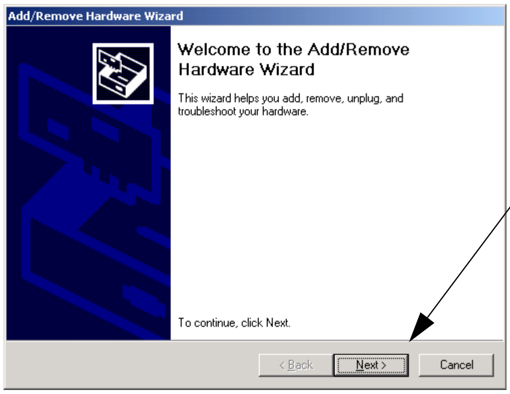
Control Panel

Add/Remove Hardware
Installs, removes, and troubleshoots hardware

Installs, removes, and troubleshoots hardware

Double-click the "Add/ Remove Hardware" icon.

3 Click the **Next** button in the **Add/Remove Hardware Wizard** window.



Welcome to the Add/Remove Hardware Wizard

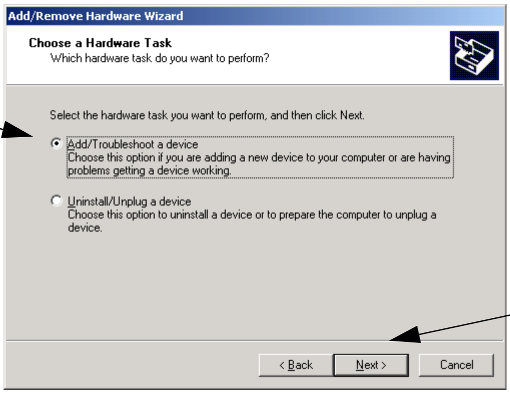
This wizard helps you add, remove, unplug, and troubleshoot your hardware.

To continue, click Next.

Click "Next" when ready.

4 Click the **Next** button.

5 Select the **Add/Troubleshoot a device** radio button in the **Add/Remove Hardware Wizard** window.



Choose a Hardware Task
Which hardware task do you want to perform?

Select the hardware task you want to perform, and then click Next.

● Add/Troubleshoot a device
Choose this option if you are adding a new device to your computer or are having problems getting a device working.

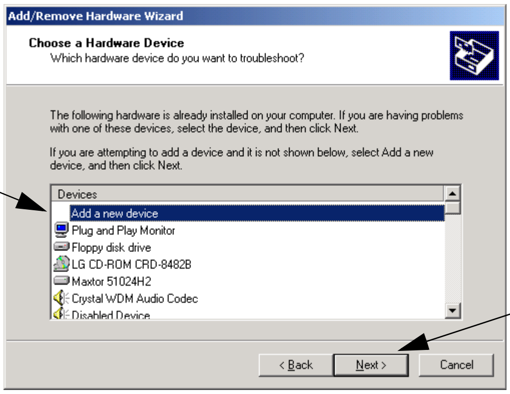
○ Uninstall/Unplug a device
Choose this option to uninstall a device or to prepare the computer to unplug a device.

Select the "Add/Troubleshoot a device" radio button.

Click "Next" when done.

6 Click the **Next** button.

7 Select **Add a new device** in the **Choose a Hardware Device** window.



Add/Remove Hardware Wizard

Choose a Hardware Device
Which hardware device do you want to troubleshoot?

The following hardware is already installed on your computer. If you are having problems with one of these devices, select the device, and then click Next.

If you are attempting to add a device and it is not shown below, select Add a new device, and then click Next.

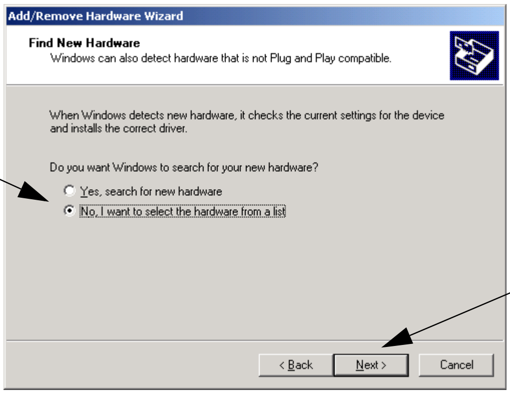
Devices
- Add a new device
- Plug and Play Monitor
- Floppy disk drive
- LG CD-ROM CRD-8482B
- Maxtor 51024H2
- Crystal WDM Audio Codec
- Disabled Device

Select “Add a new device.”

Click “Next” when done.

8 Click the **Next** button.

9 Select the **No** option radio button in the **Find New Hardware** window.



Add/Remove Hardware Wizard

Find New Hardware
Windows can also detect hardware that is not Plug and Play compatible.

When Windows detects new hardware, it checks the current settings for the device and installs the correct driver.

Do you want Windows to search for your new hardware?

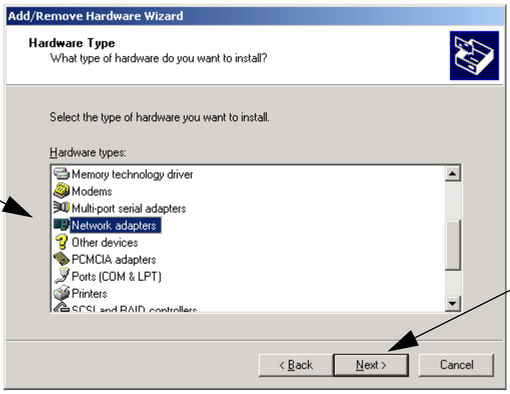
○ Yes, search for new hardware
● No, I want to select the hardware from a list

Select “No” option radio button.

Click “Next” when done.

10 Click the **Next** button.

11 Select the **Network adapters** option in the **Hardware Type** window.



Add/Remove Hardware Wizard

Hardware Type
What type of hardware do you want to install?

Select the type of hardware you want to install.

Hardware types:
- Memory technology driver
- Modems
- Multi-port serial adapters
- Network adapters
- Other devices
- PCMCIA adapters
- Ports (COM & LPT)
- Printers
- SCSI and RAID controllers

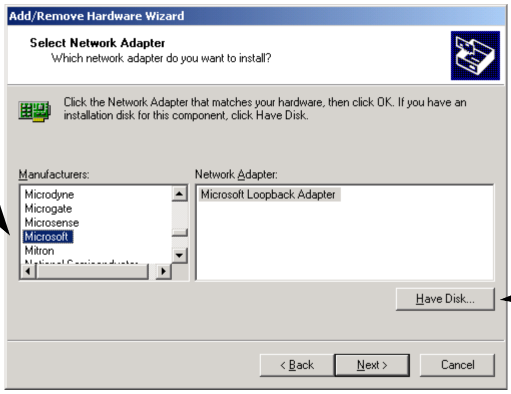
Select Network Adapters.”

Click “Next” when done.

12 Click the **Next** button.

13 Select **Microsoft** in the **Manufacturers** window.

If the Microsoft loopback adapter has been installed it will be listed in the **Network Adapter** window as shown in the figure below. If this adapter is listed, proceed to Step 17 on page 27-20. Otherwise, proceed to Step 14.



Add/Remove Hardware Wizard

Select Network Adapter
Which network adapter do you want to install?

Click the Network Adapter that matches your hardware, then click OK. If you have an installation disk for this component, click Have Disk.

Manufacturers:
Microdyne
Microgate
Microsense
Microsoft
Mitron

Network Adapter:
Microsoft Loopback Adapter

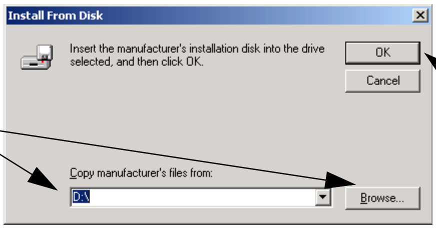
Select "Microsoft."

Click "Have Disk..." when done.

14 Click the **Have Disk...** button.

15 Load the CD or floppy disc with the Microsoft loopback adapter.

16 If needed, enter the file and path name of the Microsoft loopback adapter or click the **Browse...** button to search for the file.



Install From Disk

Insert the manufacturer's installation disk into the drive selected, and then click OK.

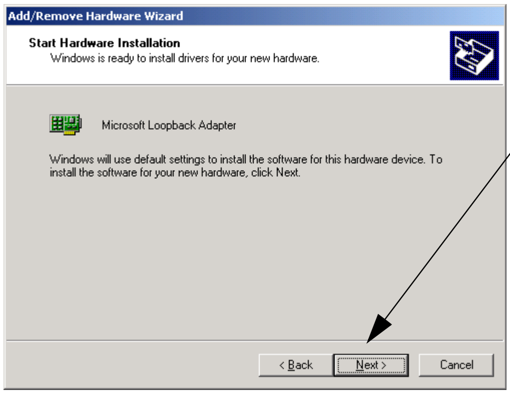
Copy manufacturer's files from:
D:\

Enter the file and path name or click the "Browse..." button to search for the file.

Click "OK" when done.

17 Click the **Next** button in the **Select Network Adapter** window (see the figure in Step 13 on page 27-20).

18 Click the **Next** button (this will install all default values) in the **Start Hardware Installation** window.



Add/Remove Hardware Wizard

Start Hardware Installation
Windows is ready to install drivers for your new hardware.

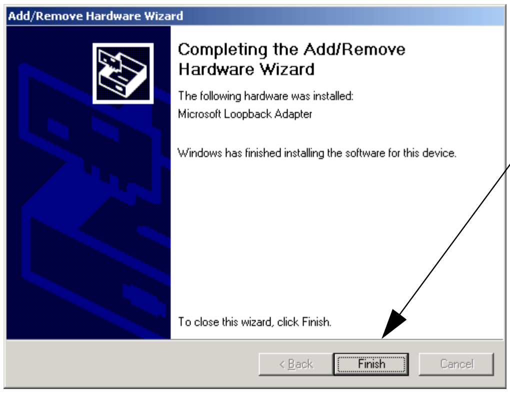
Microsoft Loopback Adapter

Windows will use default settings to install the software for this hardware device. To install the software for your new hardware, click Next.

Click "Next" when ready.

19 Click the **Next** button.

20 Click the **Finish** button in the **Completing the Add/Remove Hardware Wizard** window.



Completing the Add/Remove Hardware Wizard

The following hardware was installed:
Microsoft Loopback Adapter

Windows has finished installing the software for this device.

To close this wizard, click Finish.

Click “Finish” when ready.

Configuring a Loopback Interface on Unix- and Linux-Based Servers

This section describes how to configure a loopback interface on Red Hat Linux servers (see "Configuring a Red Hat Linux Server" on page 27-22), Sun Solaris servers (see "Configuring a Sun Solaris Server" on page 27-22) and IBM AIX servers (see "Configuring an IBM AIX Server" on page 27-23).

Note. For other versions of the Unix and Linux operating systems, please refer to your user documentation.

Configuring a Red Hat Linux Server

Follow the steps below to configure the loopback interface on a Red Hat Linux server.

1 At the command prompt, enter **ifconfig lo:1**, the Virtual IP (VIP) address of the Server Load Balancing (SLB) cluster, **netmask**, the net mask for the VIP, and **up**. For example, to configure the loopback address on a Red Hat Linux server with a VIP of 10.123.11.14 with a net mask of 255.0.0.0 enter:

```
ifconfig lo:1 10.123.11.14 netmask 255.0.0.0 up
```

2 If you do not have a file with local host names, create one. In this example we will use create a file called "/etc/localhosts.cw".

3 Add a user-configured name for the loopback interface to the file created in Step 2. In this example we will use "loopbackVIP".

4 Create a line in the **/etc/hosts** file with the VIP you configured in Step 1, a space, and the name you configured in Step 2. For example, if the VIP address is 10.123.11.14 and the name you configured in the **/etc/localhosts.cw** file is "loopbackVIP", add the following line to the **/etc/hosts** file:

```
10.123.11.14 loopbackVIP
```

Configuring a Sun Solaris Server

Follow the steps below to configure the loopback interface on a Sun Solaris server:

1 At the command prompt, enter **ifconfig lo0:1**, the Virtual IP (VIP) address of the Server Load Balancing (SLB) cluster, **netmask**, the net mask for the VIP, and **up**. For example, to configure the loopback address on a Sun Solaris server with a VIP of 10.123.11.14 with a net mask of 255.0.0.0 enter:

```
ifconfig lo0:1 10.123.11.14 255.0.0.0 up
```

2 Create a file called "/etc/hostname.lo0:1" file with the user configured name for the loopback interface. In this example we will use "loopbackVIP".

3 Create a line in the **/etc/hosts** file with the VIP you configured in Step 1, a space, and the name you configured in Step 2. For example, if the VIP address is 10.123.11.14 and the name you configured in the **/etc/hostname.lo0:1** file is "loopbackVIP", add the following line to the **/etc/hosts** file:

```
10.123.11.14 loopbackVIP
```

Configuring an IBM AIX Server

Follow the steps below to configure the loopback interface on an IBM AIX server.

1 At the command prompt, enter **ifconfig lo0 alias**, the Virtual IP (VIP) address of the Server Load Balancing (SLB) cluster, **netmask**, and the net mask for the VIP. For example, to configure the loopback address on a IBM AIX server with a VIP of 10.123.11.14 with a net mask of 255.0.0.0 enter:

```
ifconfig lo0 alias 10.123.11.14 netmask 255.0.0.0
```

2 Create a file called "/etc/hostname.lo0:1" file with the user configured name for the loopback interface. In this example we will use "loopbackVIP".

3 Create a line in the **/etc/hosts** file with the VIP you configured in Step 1, a space, and the name you configured in Step 2. For example, if the VIP address is 10.123.11.14 and the name you configured in the **/etc/hostname.lo0:1** file is "loopbackVIP", add the following line to the **/etc/hosts** file:

```
10.123.11.14 loopbackVIP
```

Configuring a Virtual IP Address on a Novell Netware 6 Server

Follow the steps below to configure the VIP (i.e., secondary) address on a Novell Netware 6 server.

Note. For other versions of Netware, please refer to your server documentation.

1 At the server prompt enter **add secondary ipaddress** followed by the VIP address and **noarp**. For example, to configure a VIP address of 10.123.11.14 enter:

```
add secondary IPAddress 10.123.11.14 noarp
```

Note. As a option you can enter **prompt** (which allows you to select from available interfaces) after the **noarp** keyword. If you do not use the **prompt** keyword then the VIP will be added to the first bound interface of the same network.

2 As an option you can enter display **secondary ipaddress** at the server prompt to verify your VIP address.

Note. If you wish to delete a VIP address enter **del secondary ipaddress** followed by the VIP address.

Configuring Server Load Balancing on a Switch

This section describes how to use Alcatel's Command Line Interface (CLI) commands to configure Server Load Balancing (SLB) on a switch.

Note. See "Quick Steps for Configuring Server Load Balancing (SLB)" on page 27-4 for a brief tutorial on configuring these mandatory parameters.

When configuring SLB parameters for an SLB cluster, you must perform the following steps:

1 Enable Server Load Balancing on Your Switch. To enable Server Load Balancing (SLB) on a switch, use the **ip slb admin** command, which is described in "Enabling and Disabling Server Load Balancing" on page 27-24.

2 Configure the Logical Server Load Balancing Cluster. To configure a logical SLB cluster, use the **ip slb cluster** command, which is described in "Configuring and Deleting SLB Clusters" on page 27-25.

3 Assign Physical Servers to the Logical Server Load Balancing Cluster. To add physical servers to a logical SLB cluster, use the **ip slb server ip cluster** command, which is described in "Assigning Servers to and Removing Servers from a Cluster" on page 27-26.

Note. Routing (which is enabled by default) must be enabled on a switch or Server Load Balancing will not operate.

Alcatel SLB software is preconfigured with the default values shown in the table in "Server Load Balancing Default Values" on page 27-3. Depending on the requirements of your network and server farm, you may need to configure more parameters than the mandatory ones described in this section. See "Modifying Optional Parameters" on page 27-28 for informational on configuring additional SLB parameters.

Enabling and Disabling Server Load Balancing

By default, Server Load Balancing (SLB) is disabled on a switch. The following subsections describe how to enable and disable SLB on a switch with the **ip slb admin** command.

Note. You must enable or disable Server Load Balancing on an entire switch. You cannot enable SLB on a per port or per slot basis.

Enabling SLB

To enable SLB switch-wide, use the **ip slb admin** command by entering:

```
-> ip slb admin enable
```

Disabling SLB

To disable SLB switch wide, use the **ip slb admin** command by entering:

```
-> ip slb admin disable
```

Configuring and Deleting SLB Clusters

The following subsections describe how to configure and delete SLB clusters with the **ip slb cluster** command.

Note. In the current release you can configure up to 15 SLB clusters on an OmniSwitch 7700/7800/8800 switch.

Configuring an SLB Cluster

You configure a logical SLB cluster with the **ip slb cluster** command by entering **ip slb cluster**, the user-specified name of the SLB cluster (which must be at least 1 character and less than or equal to 23 characters long), **vip**, and the user specified Virtual IP (VIP) address of the SLB cluster in dotted decimal notation.

Note. The VIP address of the SLB cluster *must* be an address in the same subnet as the servers.

For example, to configure an SLB cluster called “Web_Server” with a VIP address of 10.123.11.14 you would enter:

```
-> ip slb cluster Web_Server vip 10.123.11.14
```

Note. If you want spaces in an SLB cluster name, the entire name must be enclosed within quotation marks (e.g., “web server”).

When you configure an SLB cluster a Quality of Service (QoS) policy condition, action, and rule are automatically configured for it. In addition, the switch software automatically names the condition, action, and rule by adding the prefix **SLB-cond-**, **SLB-act-**, and **SLB-rule-**, respectively, to the name of the SLB cluster for each name.

For example, if you configured an SLB cluster called “Web_Server” a policy condition called “SLB-cond-Web_Server,” a policy action called “SLB-act-Web_Server,” and a policy rule called “SLB-rule-Web_Server” would be created.

You can display QoS policy rules with the **show policy rule** command. To use this command, enter **show policy rule** followed by the name of the rule. For example, to display a policy rule called “SLB-rule-Web_Server” enter:

```
-> show policy rule SLB-rule-Web_Server
```

A screen similar to the following will be displayed:

```
    Policy              From    Prec   Enabled  Inactive Reflexive  Save
+SLB-rule-Web_Server   api     0      Yes      No       No         Yes
    Cnd/Act:            SLB-cond-Web_Server -> SLB-act-Web_Server
```

You can also use the **show policy condition** to display policy conditions and the **show policy action** command to display policy actions. See Chapter 24, “Configuring QoS,” for more information on configuring and displaying QoS.

Deleting an SLB Cluster

To delete an SLB cluster, use the **no** form of the **ip slb cluster** command by entering **no ip slb cluster** followed by the name of the cluster.

For example, to delete an SLB called "Web_Server" you would enter:

```
-> no ip slb cluster Web_Server
```

Note. When you delete an SLB cluster you also delete the QoS policy, condition, and rule associated with it.

Assigning Servers to and Removing Servers from a Cluster

The following subsections describe how to assign servers to an SLB cluster and how to remove servers from an SLB cluster with the **ip slb server ip cluster** command.

Note. You can also use the **ip slb server ip cluster** command to administratively disable or enable a server (see "Taking a Server On/Off Line" on page 27-31). And you can also use this command to modify the relative weight of a server (see "Modifying the Relative Weight of a Physical Server" on page 27-30) in an SLB cluster when you add a server to a cluster or you can modify the relative weight of a server that already belongs to a cluster.

Assigning a Server to an SLB Cluster

You assign physical servers to an existing logical SLB cluster with the **ip slb server ip cluster** command by entering **ip slb server ip**, the IP address of the server in dotted decimal format, **cluster**, and the name of the SLB cluster.

For example, to assign a server with an IP address of 10.105.16.118 to an SLB cluster called "Web_Server" you would enter:

```
-> ip slb server ip 10.105.16.118 cluster Web_Server
```

You can assign up to 75 physical servers on an OmniSwitch 7700/7800/8800. For example, to assign three physical servers with IP addresses of 10.105.16.121, 10.105.16.122, and 10.105.16.123, respectively, to an SLB cluster called "Web_Server," enter the following CLI commands:

```
-> ip slb server ip 10.105.16.121 cluster Web_Server
-> ip slb server ip 10.105.16.122 cluster Web_Server
-> ip slb server ip 10.105.16.123 cluster Web_Server
```

Removing a Server from an SLB Cluster

To remove a physical server from an SLB cluster, use the **no** form of the **ip slb server ip cluster** command by entering **no ip slb server ip**, the IP address of the server you want to remove in dotted decimal format, **cluster**, and the name of the SLB cluster.

For example, to remove a server with an IP address of 10.105.16.121 from an SLB cluster called "Web_Server" you would enter:

```
-> no ip slb server ip 10.105.16.121 cluster Web_Server
```

Modifying Optional Parameters

As shown in the table on page 27-3, Alcatel's SLB software is preconfigured with default values for the SLB cluster's distribution algorithm, "sticky" time, ping timeout, ping period, and ping retries. The following subsections describe how to modify these parameters.

- **Modifying the Distribution Algorithm**. You can modify the distribution algorithm with the **ip slb cluster distribution** command, which is described in "Modifying the Distribution Algorithm" on page 27-28.
- **Modifying the Sticky Time**. You can modify the sticky time with the **ip slb cluster sticky time** command, which is described in "Modifying the Sticky Time" on page 27-29.
- **Modifying the Ping Period**. You can modify the ping period with the **ip slb cluster ping period** command, which is described in "Modifying the Ping Period" on page 27-29.
- **Modifying the Ping Timeout**. You can modify the ping timeout with the **ip slb cluster ping timeout** command, which is described in "Modifying the Ping Timeout" on page 27-29.
- **Modifying the Number of Ping Retries**. You can modify the number of ping retries with the **ip slb cluster ping retries** command, which is described in "Modifying the Ping Retries" on page 27-30.
- **Modifying the Relative Weight of a Physical Server**. You can modify the relative weight of a server within an SLB cluster with the **ip slb server ip cluster** command, which is described in "Modifying the Relative Weight of a Physical Server" on page 27-30.

Modifying the Distribution Algorithm

The default distribution algorithm is *weighted round robin*, where the SLB cluster "loops" (i.e., distributes traffic) on in-service servers, taking into account their respective administrative weight. To change the distribution to *server failover*, where the SLB cluster selects the highest weighted in-service server (in the ordered list of servers) in a pool of servers, use the **ip slb cluster distribution** command by entering **ip slb cluster**, followed by the name of the cluster, and **distribution server failover**. For example:

```
-> ip slb cluster Web_Server distribution server failover
```

Note. You can use the **show ip slb cluster** command to display the ordered list (by IP address) of servers within an SLB cluster. See page 27-4 for a sample of the **show ip slb cluster** command output.

To change the distribution to weighted round robin, use the **ip slb cluster distribution** command by entering **ip slb cluster**, followed by the name of the cluster, and **distribution round robin**. For example:

```
-> ip slb cluster Web_Server distribution round robin
```

Modifying the Sticky Time

The default "sticky" time (i.e., a client is kept attached to a physical server while there is no activity) is 1200 seconds. You can modify this value from 0 (i.e., the client will always be attached to this server) to 86400 seconds with the **ip slb cluster sticky time** command by entering **ip slb cluster**, the name of the SLB cluster, **sticky time**, and the user-specified number of seconds.

For example, to set the sticky time on an SLB cluster called "Web_Server" to 600 seconds enter:

```
-> ip slb cluster Web_Server sticky time 600
```

Although you can set the sticky time to any integer between 0 and 86400, in the current release actual values are determined by the aging period for the switch's Hardware Routing Engine (HRE). On the OmniSwitch 7700/7800/8800, aging for HRE entries is done every 30 seconds. Therefore, when you configure a sticky time it will always be rounded up to a multiple of 30 seconds.

For example, configuring a sticky time of 1, 8, 15, 26, or 30 seconds will have the same result (i.e., it will be 30 seconds). Similarly, configuring a sticky time of 31, 38, 45, 56, or 60 seconds will have the same result.

Modifying the Ping Period

The default ping period (i.e., the time interval at which the health of servers is checked) is 60 seconds. You can modify this value from 0 (this will disable the ping) to 3600 seconds with the **ip slb cluster ping period** command by entering **ip slb cluster**, the name of the SLB cluster, **ping period**, and the user-specified number of seconds.

For example, to set the ping period on an SLB cluster called "Web_Server" to 1200 seconds enter:

```
-> ip slb cluster Web_Server ping period 120
```

Note. If you set the ping period to any value other than 0, then ping period must be greater than or equal to the ping timeout value divided by 1000. For example, if the ping timeout is 5000 milliseconds, the ping period must be at least 5 seconds. The ping timeout value can be modified with the **ip slb cluster ping timeout** command, which is described in "Modifying the Ping Timeout" on page 27-29.

Modifying the Ping Timeout

The default ping timeout is 3000 milliseconds. You can modify this value from 0 to 1000 times the value of the ping period with the **ip slb cluster ping timeout** command by entering **ip slb cluster**, the name of the SLB cluster, **ping timeout**, and the user-specified number of milliseconds.

For example to set the ping timeout on an SLB cluster called "Web_Server" to 1000 milliseconds enter:

```
-> ip slb cluster Web_Server ping timeout 1000
```

Note. You can modify the ping period with the **ip slb cluster ping period** command, which is described in "Modifying the Ping Period" on page 27-29.

Modifying the Ping Retries

The default number of ping retries is 3. You can modify this value from 0 to 255 with the **ip slb cluster ping retries** command by entering **ip slb cluster**, the name of the SLB cluster, **ping retries**, and the user-specified number of ping retries. For example:

```
-> ip slb cluster Web_Server ping retries 5
```

Modifying the Relative Weight of a Physical Server

You can use the **ip slb server ip cluster** command to modify the relative weight of a server in an SLB cluster from by entering **ip slb server**, the IP address of the physical server you want to modify, **cluster**, the name of the SLB cluster to which this server belongs, **weight**, and the value for the relative weight, which can range from 0 (the switch will prevent this server from being assigned any new connections) to 100, with 100 having the greatest relative weight. (The default is 10.)

For example, to set the relative weight of server with an IP address of 10.105.16.121 that belongs to an SLB cluster called "Web_Server" to 5 enter:

```
-> ip slb server ip 10.105.16.121 cluster Web_Server weight 5
```

Administrative weights are relative. For example, say Servers A and B have respective weights of 10 and 20 within a cluster. In this example, Server A would get half the traffic of server B. Since administrative weights are relative, assigning Servers A and B respective weights of 1 and 2, 5 and 10, or 25 and 50, etc., would produce identical results.

Note. The **ip slb server ip cluster** command is also used to add or remove servers from an SLB cluster (see "Assigning Servers to and Removing Servers from a Cluster" on page 27-26) and for administratively enabling and disabling a server in an SLB cluster (see "Taking a Server On/Off Line" on page 27-31).

Taking Clusters and Servers On/Off Line

Alcatel's Server Load Balancing (SLB) **show** commands provide tools to monitor traffic and troubleshoot problems. These commands are described in "Displaying Server Load Balancing Status and Statistics" on page 27-37. If problems are identified, you can use the **ip slb cluster admin status** command to administratively disable an entire SLB cluster or the **ip slb server ip cluster** command to administratively disable individual servers within an SLB cluster. These commands are described in the following sections.

Taking a Cluster On/Off Line

The following subsections describe how to bring an SLB cluster on line and how to take it off line with the **ip slb cluster admin status** command.

Bringing an SLB Cluster On Line

You can bring an administratively disabled SLB cluster on line with the **ip slb cluster admin status** command by entering **ip slb cluster**, the name of the SLB cluster, and **admin status enable**.

For example, to bring an SLB cluster called "WorldWideWeb" on line you would enter:

```
-> ip slb cluster WorldWideWeb admin status enable
```

Taking an SLB Cluster Off Line

You can take a Server Load Balancing (SLB) cluster off line with the **ip slb cluster admin status** command by entering **ip slb cluster**, the name of the SLB cluster, and **admin status disable**.

For example, to take an SLB cluster called "WorldWideWeb" off line you would enter:

```
-> ip slb cluster WorldWideWeb admin status disable
```

Taking a Server On/Off Line

The following subsections describe how to bring a physical server on line and how to take it off line with the **ip slb server ip cluster** command.

Note. The **ip slb server ip cluster** command is also used to add or remove physical servers from an SLB cluster (see "Assigning Servers to and Removing Servers from a Cluster" on page 27-26) and for modifying the relative weight of a server in an SLB cluster (see "Modifying the Relative Weight of a Physical Server" on page 27-30).

Bringing a Server On Line

You bring an administratively disabled server in an SLB cluster on line with the **ip slb server ip cluster** command by entering **ip slb server**, the IP address of the server you want to enable in dotted decimal format, **cluster**, the name of the SLB cluster to which the server belongs, and **admin status enable**.

For example, to administratively enable a server with an IP address of 10.105.16.121 that belongs to an SLB cluster called "Web_Server" you would enter:

```
-> ip slb server ip 10.105.16.121 cluster Web_Server admin status enable
```

Taking a Server Off Line

You can administratively disable a server in an SLB cluster and take it off line with the **ip slb server ip cluster** command by entering **ip slb server**, the IP address of the server you want to disable in dotted decimal format, **cluster**, the name of the SLB cluster to which the server belongs, and **admin status disable**.

For example, to administratively disable a server with an IP address of 10.105.16.123 that belongs to an SLB cluster called "Web_Server" you would enter:

```
-> ip slb server ip 10.105.16.123 cluster Web_Server admin status disable
```

Configuring SLB Probes

Server Load Balancing (SLB) probes allow you to check the health of logical clusters and physical servers. Supported features include:

- Support for server health monitoring using Ethernet link state detection
- Support for server health monitoring using IPv4 ICMP ping
- Support for server health monitoring using a Content Verification Probe.

Creating SLB Probes

To create an SLB probe use the **ip slb probe** command by entering **ip slb probe** followed by the user-configured probe name and the probe type, which can be any one of the following listed in the table below:

ip slb probe keywords		
ftp	**http**	**https**
imap	**imaps**	**nntp**
ping	**pop**	**pops**
smtp	**tcp**	**udp**

For example, to create an HTTP SLB probe called “server_probe1” enter:

```
-> ip slb probe server_probe1 http
```

You can configure up to 20 probes on a switch.

Deleting SLB Probes

To delete an SLB use the **no** form of the **ip slb probe** command by entering **no ip slb probe** followed by the probe name. For example, to delete an SLB probe called “server_probe1” enter:

```
-> no ip slb probe server_probe1
```

Associating a Probe with a Cluster

To associate an existing SLB probe with a cluster use the **ip slb cluster probe** command by entering **ip slb cluster** followed by the user-configured cluster name, **probe**, and the user-configured probe name.

For example, to associate a probe called “cluster_probe1” with a cluster called “WorldWideWeb” enter:

```
-> ip slb cluster WorldWideWeb probe cluster_probe1
```

Associating a Probe with a Server

To associate an existing SLB probe with a server use the **ip slb server ip cluster probe** command by entering **ip slb server ip** followed by IP address of the server, **cluster**, the user-configured cluster name, **probe**, and the user-configured probe name.

For example, to associate a probe called "server_probe1" with a server with an IP address of 10.255.11.127 that belongs to a cluster called "WorldWideWeb" enter:

```
-> ip slb server ip 10.255.11.127 cluster WorldWideWeb probe server_probe1
```

Modifying SLB Probes

The following subsections describe how to modify existing SLB probes.

Modifying the Probe Timeout

By default, the timeout used to wait for SLB probe answers is 3000 seconds. To modify this value from 0 to 3600000 seconds use the **ip slb probe timeout** command by entering **ip slb probe** followed by the user-configured probe name, the probe type, **timeout**, and the user-specified timeout value.

Note. See "Creating SLB Probes" on page 27-33 for a list of valid probe types.

For example, to set the timeout for an HTTP SLB probe called "server_probe1" to 12000 seconds enter:

```
-> ip slb probe server_probe1 http timeout 12000
```

Modifying the Probe Period

By default, the SLB probe period to check the health of servers is 60 seconds. To modify this value from 0 to 3600 seconds use the **ip slb probe period** command by entering **ip slb probe** followed by the user-configured probe name, the probe type, **period**, and the user-specified period value.

Note. See "Creating SLB Probes" on page 27-33 for a list of valid probe types.

For example, to set the period for an HTTP SLB probe called "server_probe1" to 120 seconds enter:

```
-> ip slb probe server_probe1 http period 120
```

Modifying the Probe TCP/UDP Port

By default, the TCP/UDP port the SLB probe should be sent on is 0. To modify this value from 0 to 65535 use the **ip slb probe port** command by entering **ip slb probe** followed by the user-configured probe name, the probe type, **port**, and the user-specified port number.

Note. See "Creating SLB Probes" on page 27-33 for a list of valid probe types.

For example, to set the TCP/UDP port for an HTTP SLB probe called "server_probe1" to 200 enter:

```
-> ip slb probe server_probe1 http port 200
```

Modifying the Probe Retries

By default, the number of SLB probe retries before deciding that a server is out of service is 3. To modify this value from 0 to 255 use the **ip slb probe retries** command by entering **ip slb probe** followed by the user-configured probe name, the probe type, **retries**, and the user-specified number of retries.

Note. See "Creating SLB Probes" on page 27-33 for a list of valid probe types.

For example, to set the number of retries for an HTTP SLB probe called "server_probe1" to 10 enter:

```
-> ip slb probe server_probe1 http retries 10
```

Configuring a Probe User Name

To configure a user name sent to a server as credentials for an HTTP GET operation to verify the health of the server use the **ip slb probe username** command by entering **ip slb probe** followed by the user-configured probe name, either **http** or **https**, **username**, and the user-specified user name.

For example, to set the user name for an HTTP SLB probe called "server_probe1" to "subnet1" enter:

```
-> ip slb probe server_probe1 http username subnet1
```

Configuring a Probe Password

To configure a password sent to a server as credentials for an HTTP GET to verify the health of the server use the **ip slb probe password** command by entering **ip slb probe** followed by the user-configured probe name, either **http** or **https**, **password**, and the user-specified password.

For example, to set the password for an HTTP SLB probe called "server_probe1" to "h1f45xc" enter:

```
-> ip slb probe server_probe1 http password h1f45xc
```

Configuring a Probe URL

To configure a URL sent to a server for an HTTP GET to verify the health of the server use the **ip slb probe url** command by entering **ip slb probe** followed by the user-configured probe name, either **http** or **https**, **url**, and the user-specified URL.

Note. The URL should be the relative web page name to be retrieved.

For example, to set the URL for an HTTP SLB probe called "server_probe1" to "pub/index.html" enter:

```
-> ip slb probe server_probe1 http url pub/index.html
```

Modifying the Probe Status

By default, the expected status returned from an HTTP GET to verify the health of a server is 200. To modify this value from 0 to 4294967295 use the **ip slb probe status** command by entering **ip slb probe** followed by the user-configured probe name, either **http** or **https**, **status**, and the user-specified expected status.

For example, to set the expected status for an HTTP SLB probe called "server_probe1" to 404 enter:

```
-> ip slb probe server_probe1 http status 404
```

Configuring a Probe Send

To configure an ASCII string sent to a server to invoke a response from it and to verify its health use the **ip slb probe send** command by entering **ip slb probe** followed by the user-configured probe name, the valid probe type (**http**, **https**, **udp**, or **tcp**), **send**, and the user-specified ASCII string.

For example, to set the TCP/UDP port for an HTTP SLB probe called “server_probe1” to “test” enter:

```
-> ip slb probe server_probe1 http send test
```

Configuring a Probe Expect

To configure an ASCII string used to compare a response from a server to verify the health of the server use the **ip slb probe expect** command by entering **ip slb probe** followed by the user-configured probe name, the valid probe type (**http**, **https**, **udp**, or **tcp**), **expect**, and the user-specified ASCII string.

For example, to set the TCP/UDP port for an HTTP SLB probe called “server_probe1” to “test” enter:

```
-> ip slb probe server_probe1 http expect test
```

Displaying Server Load Balancing Status and Statistics

You can use CLI **show** commands to display the current configuration and statistics of Server Load Balancing on a switch. These commands include the following:

show ip slb Displays the status of server load balancing on a switch.

show ip slb servers Displays the status of all physical servers belonging to server load balancing clusters on a switch.

show ip slb clusters Displays the status and configuration of all server load balancing clusters on a switch.

show ip slb cluster Displays detailed status and configuration information for a single server load balancing cluster on a switch.

show ip slb cluster server Displays detailed status and configuration information for a single physical server in a server load balancing cluster.

show ip slb probes Display the configuration of Server Load Balancing (SLB) probes.

The **show ip slb**, **show ip slb servers**, and **show ip slb clusters** commands provide a “global” view of switch-wide SLB parameters. These commands are particularly helpful in fine-tuning configurations. For example, if you wanted to get a quick look at the status of all SLB clusters you would use the **show ip slb clusters** command as shown below:

```
-> show ip slb clusters
                                  Admin    Operational         #      %
Cluster Name     VIP              Status   Status              Srv    Avail
----------------+----------------+--------+-------------------+-----+---------
WorldWideWeb     128.241.130.204  Enabled  In Service          3      95
Intranet         128.241.130.205  Enabled  In Service          2      100
FileTransfer     128.241.130.206  Enabled  Out of Service      2      50
```

In the example above, two SLB clusters (“WorldWideWeb” and “Intranet”) are administratively enabled and are “in service” (i.e., at least one physical server is operational in the cluster). The third SLB cluster (“FileTransfer”) is administratively enabled but is “out of service (i.e.,no physical servers are operational in the cluster).

The **show ip slb cluster** and **show ip slb cluster server** commands provide detailed configuration information and statistics for individual SLB clusters and servers, respectively. To use the **show ip slb cluster** command, enter **show ip slb cluster** followed by the name of the SLB cluster, as shown below:

```
-> show ip slb cluster Web_Server
```

Note. See page 27-4 for a sample of the **show ip slb cluster** command output.

To use the **show ip slb cluster server** command, enter **ip slb cluster**, the name of the SLB cluster that the server belongs to, **server**, and the IP address of the server. For example, to display statistics and parameters for a server with an IP address of 10.123.11.14 that belongs to an SLB cluster called “Web_Server” you would enter:

```
-> show ip slb cluster Web_Server server 10.123.11.14
```

A screen similar to the following will be displayed:

```
Cluster Web_Server
VIP: 10.123.11.14
   Server 10.123.11.4
   Admin weight                            = 10,
   MAC addr                                : 00:00:1f:40:53:6a,
   Slot number                             = 1,
   Port number                             = 4,
   Admin status                            : Enabled,
   Operational status                      : In Service,
   Availability time (%)                   = 95,
   Ping failures                           = 0,
   Last ping round trip time (milliseconds)= 20,
   Link down count                         = 1,
   Number of flows                         = 2000
```

In the example above, the server with an IP address of 10.123.11.4 is shown to be administratively enabled and “in service” (i.e., this server is being used for SLB cluster client connections). In addition, operational parameters (e.g., link down count) are also displayed.

The **show ip slb probes** command provides both a global view of SLB probes and a detailed configuration information and statistics for individual probes. For example, to view the status of all probes enter show ip slb probes as shown below:

```
-> show ip slb probes
Probe Name              Period  Retries Timeout Method
-----------------------+-------+-------+--------+------
web_server                60000       3   12000   HTTP
mail_server               60000       3    3000   SMTP
mis_servers             3600000       5   24000   Ping
```

In the example above there are three probes configured on the switch.

To view detailed information on a single probe enter **show ip slb probes** followed by the probe name as shown in the example below:

```
-> show ip slb probes phttp
Probe phttp
   Type                           = HTTP,
   Period (seconds)               = 60,
   Timeout (milliseconds)         = 3000,
   Retries                        = 3,
   Port                           = 0,
   Username                       = ,
   Password                       = ,
   Expect                         = ,
   Status                         = 200,
   URL                            = /,
```

Note. See the “Server Load Balancing Commands” chapter in the *OmniSwitch CLI Reference Guide* for complete syntax information on SLB **show** commands.

28 Configuring High Availability VLANs

High availability (HA) VLANs, unlike standard VLANs, allow you to send traffic intended for a single destination MAC address to multiple switch ports. These high availability VLANs can be used to manage server clusters.

In This Chapter

This chapter describes the basic components of high availability VLANs and how to configure them through the Command Line Interface (CLI). CLI commands are used in the configuration examples; for more details about the syntax of commands, see the *OmniSwitch CLI Reference Guide*.

Configuration procedures described in this chapter include:

- Creating a VLAN on page 28-9.
- Adding ingress ports to a high availability VLAN on page 28-10.
- Adding egress ports to a high availability VLAN on page 28-12.
- Adding MAC addresses to a high availability VLAN on page 28-13.
- Configuring inter-switch ports for HA VLANs on page 28-14.
- Configuring the ingress flood queue bandwidth on page 28-15

Note. You can also configure and monitor high availability VLANs with WebView, Alcatel's embedded web-based device management application. WebView is an interactive and easy-to-use GUI that can be launched from OmniVista or a web browser. Please refer to WebView's online documentation for more information on configuring and monitoring high availability VLANs with WebView.

High Availability VLANs Specifications

The table below lists specifications for high availability VLAN software.

RFCs Supported	2674—*Definitions of Managed Objects for Bridges with Traffic Classes, Multicast Filtering and Virtual LAN Extensions*
IEEE Standards Supported	802.1D—*Media Access Control Bridges* 802.1w—*Rapid Reconfiguration (802.1D Amendment 2*) 802.1s—*Multiple Spanning Trees (802.1Q Amendment 3)*
Maximum high availability VLANs per switch	32
Switch ports eligible for high availability VLAN assignment.	Fixed ports on second-generation Network Interface (NI) modules.
Switch port *not* eligible for high availability VLAN assignment.	Mobile, 802.1Q tagged, link aggregate, Learned Port Security (LPS), mirrored or mirroring ports, and all ports on first-generation NI modules.
Spanning Tree modes supported.	Flat (one Spanning Tree instance per switch). 1x1 (one Spanning Tree instance per VLAN).
CLI Command Prefix Recognition	All high availability VLAN configuration commands with the **vlan** prefix support prefix recognition. (However, the **show mac-address-table port-mac** command does not support prefix recognition.) See the “Using the CLI” chapter in the *OmniSwitch 7700/7800/8800 Switch Management Guide* for more information.

High Availability Default Values

The table below lists default values for high availability VLAN software.

Parameter Description	Command	Default Value/Comments
Ingress ports assigned.	**vlan port-mac ingress-port**	No ingress ports assigned.
Egress ports assigned.	**vlan port-mac egress-port**	No egress ports assigned.
MAC addresses assigned.	**mac-address-table port-mac** **vlan mac**	No MAC addresses assigned.
high availability VLAN ingress Flood queue bandwidth	**vlan port-mac bandwidth**	15 Mbps

Quick Steps for Creating High Availability VLANs

Follow the steps below for a quick tutorial on configuring high availability (HA) VLANs. Additional information on how to configure each command is given in the sections that follow.

1 Create a default VLAN for the HA VLAN ports with the **vlan** command as shown below:

```
-> vlan 10
```

2 Assign ports to the new default VLAN with the **vlan port default** command as shown below:

```
-> vlan 10 port default 1/1, 3/2, 3/7, 6/1, 6/2
```

3 Create a VLAN that will become the HA VLAN once configured with ingress and egress ports. For example:

```
-> vlan 200
```

4 Configure ports 1/1 and 3/2 as ingress ports for HA VLAN 200. Use the **vlan port-mac ingress-port** command as shown below:

```
-> vlan 200 port-mac ingress-port 1/1 3/2
```

5 Configure ports 6/1 and 6/2 as egress ports for HA VLAN 200. Use the **vlan port-mac egress-port** command as shown below:

```
-> vlan 200 port-mac egress-port 6/1 6/2
```

6 Assign a MAC address to this high availability VLAN with the **mac-address-table port-mac vlan mac** command. For example:

```
-> mac-address-table port-mac vlan 200 mac 00:DA:29:3C:11:20
```

7 Configure port 3/7 as an inter-switch port for the HA VLAN with the **vlan 802.1q** command as shown below:

```
-> vlan 200 802.1q 3/7
```

Note that Step 7 is only done when the HA VLAN is configured across two switches. The ports that provide the connection between the two switches must be tagged with the HA VLAN ID.

8 Set the HA VLAN ingress flood queue bandwidth size to 100 Mbps. Use the **vlan port-mac bandwidth** command as shown below:

```
-> vlan 200 port-mac bandwidth 100
```

Note. *Optional*. You can display the configuration of high availability VLANs with the **show mac-address-table port-mac** command. For example:

```
-> show mac-address-table port-mac 200
Port mac configuration for vlan 200

Bandwidth : 100 MB/sec

   Ingress Port list:
           1/1  3/2
   Egress Port list:
           6/1  6/2
   Mac Address list:
           00:DA:29:3C:11:20
```

An example of what these commands look like entered sequentially on the command line:

```
-> vlan 10
-> vlan 10 port default 1/1 3/2 3/7 6/1 6/2
-> vlan 200 port-mac ingress-port 1/1 3/2
-> vlan 200 port-mac egress-port 6/1 6/2
-> mac-address-table port-mac vlan 200 mac 00:DA:29:3C:11:20
-> vlan 200 802.1q 3/7
-> vlan 200 port-mac bandwidth 100
```

High Availability VLAN Overview

High availability (HA) VLANs send traffic intended for a single destination MAC address to multiple switch ports. This section provides a brief overview on how traffic flows in and out of high availability VLANs and how high availability VLANs can manage third-party high availability firewall clusters (see “High Availability Firewall Clusters” on page 28-7 for information).

An HA VLAN is configured by creating a standard VLAN and then assigning ingress or egress ports to the VLAN. Once these types of ports are assigned, the standard VLAN automatically becomes an HA VLAN. When this occurs, standard VLAN commands no longer apply.

Destination MAC addresses (unicast and multicast) are also assigned to high availability VLANs. These addresses identify ingress port traffic that the switch will send out on all egress ports that belong to the same VLAN

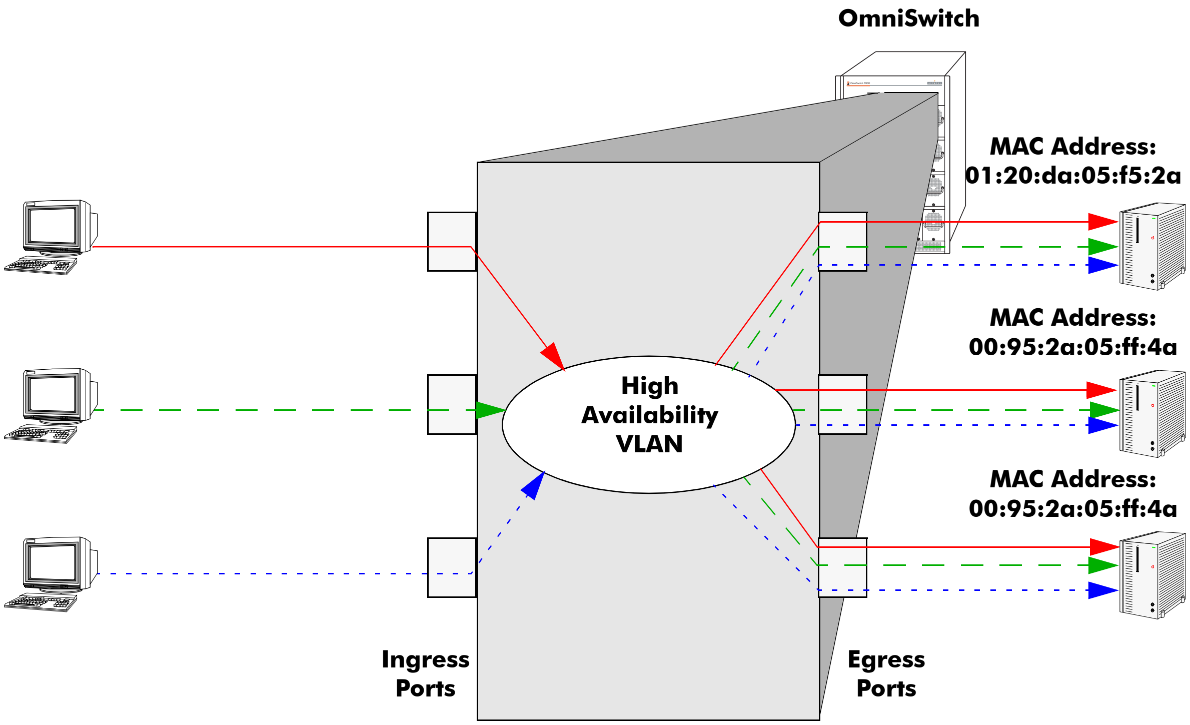
In addition to assigning ingress and egress ports, tagging inter-switch link ports with an HA VLAN ID is allowed. Ingress port traffic destined for an HA VLAN MAC address is sent out on all egress *and* inter-switch link ports that belong to the same VLAN. Traffic forwarded on inter-switch link ports is done so in accordance with the Spanning Tree state of the port.

It is also possible to configure the ingress flood queue bandwidth size for HA VLANs. See “Configuring the Flood Queue Bandwidth” on page 28-15 for more information.

Note. Once a VLAN becomes an HA VLAN, only ingress, egress, and tagged inter-switch link ports are allowed as members of that VLAN.

Ingress and Egress Traffic Flows

The figure below shows how ingress traffic is handled by high availability VLANs.



Ingress to Egress Port Flow

In the above example, packets received on the ingress ports that are destined for the high availability VLAN MAC address are sent out the egress ports that are members of the same VLAN. Since all three servers are connected to egress ports, they all receive the ingress port traffic. This provides a high level of availability in that if one of the server connections goes down, the other connections still forward traffic to one of the redundant servers.

Note the following regarding ingress and egress port traffic flow:

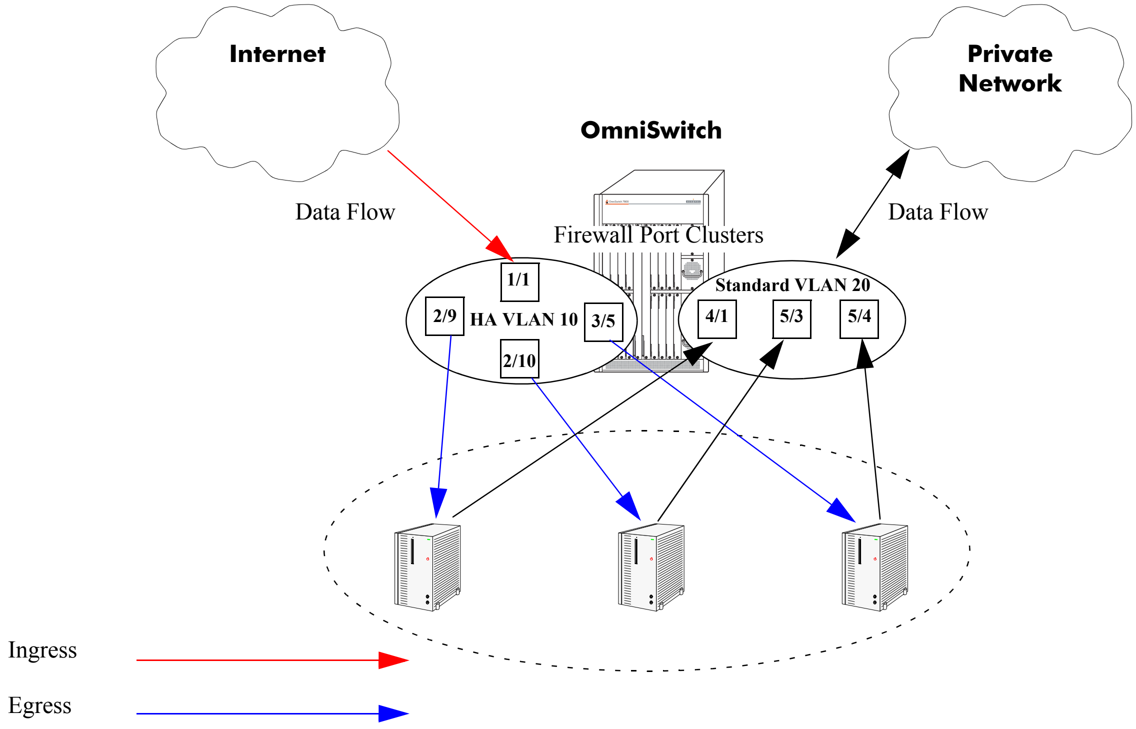
- Ingress port traffic destined for the high availability VLAN MAC address is only sent out on egress ports and not on any other ingress ports.
- If a packet received on an ingress port is not destined for the high availability VLAN MAC address, the packet is bridged as regular traffic to all ports in the VLAN, not just egress ports.
- Traffic received on egress ports is bridged as regular traffic to all ports assigned to the VLAN, regardless of their ingress or egress port state.

High Availability Firewall Clusters

One key application of high availability VLANs is interfacing with third-party high availability firewall clusters, which allow two or more servers running a common firewall application to work as if they were one system. The following subsection describes an example HA VLAN implementation that is used to interface with a third-party high availability firewall cluster.

Traditional Firewall Implementation

The figure below shows two high availability VLANs that are used to manage a third-party high availability firewall cluster. Unsecure traffic from the Internet comes into the OmniSwitch through the ingress port 1/1 of high availability VLAN 10. This traffic is sent to the high availability cluster through the egress ports that belong to HA VLAN 10 (2/9, 2/10, and 3/5).



Firewall and High Availability Cluster

The third-party high availability firewall cluster sends authorized traffic to ports 4/1, 5/3, and 5/4 that belong to standard VLAN 20 . This traffic is then forwarded on VLAN 20 to the private network.

See “Application Example 1: Firewall Cluster” on page 28-16 for instructions on how to configure the high availability VLANs in the example above.

Configuring High Availability VLANs on a Switch

This section describes how to use the Command Line Interface (CLI) commands to configure high availability (HA) VLANs on a switch. For a brief tutorial on configuring HA VLANs, see "Quick Steps for Creating High Availability VLANs" on page 28-3.

When configuring HA VLANs, you must perform the following steps:

1 **Create a VLAN**. To create a VLAN use the **vlan** command, which is described in "Creating and Deleting VLANs" on page 28-9.

2 **Assign Ingress Ports.** To assign ingress ports to the high availability VLAN, use the **vlan port-mac ingress-port** command, which is described in "Assigning and Removing Ingress Ports" on page 28-10.

3 **Assign Egress Ports.** To assign egress ports to the high availability VLAN, use the **vlan port-mac egress-port** command, which is described in "Assigning and Removing Egress Ports" on page 28-12.

4 **Assign MAC Addresses**. To assign MAC addresses to the high availability VLAN, use the **mac-address-table port-mac vlan mac** command, which is described in "Assigning and Removing MAC Addresses" on page 28-13.

5 **Configure Inter-Switch Ports.** To configure an HA VLAN across two switches, use the **vlan 802.1q** command to tag the connection ports with the HA VLAN ID. This procedure is described in "Configuring Inter-switch Ports for HA VLANs" on page 28-14.

6 **Configure The Flood Queue Bandwidth**. To configure the size of the HA VLAN ingress flood queue bandwidth, use the **vlan port-mac bandwidth** command, which is described in "Configuring the Flood Queue Bandwidth" on page 28-15.

Note. You must have write access to the VLAN family of commands (i.e., the **domain-layer2** domain) to use the commands described in the following subsections. See the "Managing Switch User Accounts" in the *OmniSwitch 7700/7800/8800 Switch Management Guide* for more information.

Note the following when configuring HA VLANs:

- Only fixed ports on second-generation Network Interface (NI) modules are eligible for HA VLAN assignment. Mobile ports, 802.1Q tagged ports, link aggregate ports, Learned Port Security (LPS) ports, and ports that mirror or are mirrored are not eligible for HA VLAN use.
- All HA VLAN related ports must first belong to the same default VLAN before they are configured as ingress, egress, or inter-switch ports for the HA VLAN.
- Only ingress/egress and tagged inter-switch ports are allowed in an HA VLAN. See "Configuring Inter-switch Ports for HA VLANs" on page 28-14 for more information about inter-switch ports.
- Do not assign an HA VLAN as the default VLAN for a port. Any attempt to do so is not allowed.
- When a port is assigned to an HA VLAN as an ingress or egress port, the default VLAN assignment remains the same. For example, if VLAN 10 is the default VLAN for port 3/10 and this same port is associated with HA VLAN 200 as an ingress or egress port, VLAN 10 still remains the default VLAN for port 3/10. In addition, the **show vlan port** command only shows the VLAN 10 assignment.
- It is possible to designate a port as both an ingress and egress port.

- It is highly recommended that all switches that participate in an HA VLAN configuration run in the same Spanning Tree mode and use the same Spanning Tree protocol (STP, RSTP, MSTP).

Use the **show mac-address-table port-mac** command to verify the HA VLAN configuration on the switch. See "Displaying High Availability VLAN Status and Statistics" on page 28-19 for more information.

Creating and Deleting VLANs

The following subsections describe how to create and delete a VLAN with the **vlan** command.

Note. This chapter provides only a basic description of creating and deleting VLANs. For a complete description of configuring and monitoring VLANs on a switch, please refer to Chapter 5, "Configuring VLANs."

Creating a VLAN

To create a new VLAN use the **vlan** command by entering **vlan** followed by the VLAN ID number, which can be any integer from 2 to 4094. (Default VLAN 1 is part of the standard switch configuration and does not need to be created.) For example, to create a VLAN with a VLAN ID number of 10 enter

```
-> vlan 10
```

You can also specify the administrative status and a name for the VLAN with the **vlan** command. For example, to administratively enable (the default) a VLAN when you configure it enter **vlan** followed by the VLAN ID number and **enable**.

For example, to create vlan 10 and administratively enable it enter

```
-> vlan 10 enable
```

To administratively disable a VLAN when you configure it enter **vlan** followed by the VLAN ID number and **disable**.

For example, to create vlan VLAN 10 and administratively disable it enter

```
-> vlan 10 disable
```

To assign a name to a VLAN when you configure it enter **vlan** followed by the VLAN ID number, **name**, and a text description, which can be up to 32 characters long.

Note. If a text description has spaces the name must be enclosed within quotes (e.g., **"VLAN 10"**)

For example, to create VLAN 10 and name it "VLAN10" enter

```
-> vlan 10 name VLAN10
```

Note. You can use the **name** keyword with the **enable** and **disable** keywords (e.g., **vlan 10 enable name VLAN10**).

Deleting a VLAN

To delete a VLAN use the **no** form of the **vlan** command by entering **no vlan** followed by the VLAN's ID number. For example, to delete high availability VLAN 10 enter:

```
-> no vlan 10
```

Assigning and Removing Ingress Ports

The following subsections describe how to assign to and remove ingress ports from a high availability VLAN with the **vlan port-mac ingress-port** command.

Note. Using the **vlan port-mac ingress-port** command will change a standard VLAN to a high availability VLAN. Standard VLAN commands do not apply to high availability VLANs.

Assigning Ingress Ports

To assign ingress ports to a high availability VLAN you use the **vlan port-mac ingress-port** command by entering **vlan**, followed by the VLAN's ID number, **port-mac ingress-port**, the slot number of the port, a slash (/), and the port number.

For example, to add ingress port 3/2 to high availability VLAN 10 you would enter:

```
-> vlan 10 port-mac ingress-port 3/2
```

You can also add multiple ingress ports by entering **vlan**, followed by the VLAN's ID number, **port-mac ingress-port**, the slot number of the first port, a slash (/), the port number of the first port, a space, the slot number of the second port, a slash, and the port number of the second port. Additional ports can also be added by specifying their slot number, a slash, and their port number preceded by a space.

Note. Ingress ports that belong to the same high availability VLAN do not need to be sequential and can be on different second-generation NI modules.

For example, to add ingress port 3/2, 5/1, and 7/4 to high availability VLAN 10 you would enter:

```
-> vlan 10 port-mac ingress-port 3/2 5/1 7/4
```

You can also add a range of ingress ports by entering **vlan**, followed by the VLAN's ID number, **port-mac ingress-port**, the slot number of the first port, a slash (/), the port number of the first port on the NI, a hyphen (**-**), and the last port on the NI.

For example, to add ingress port 2/1 through 2/5 to high availability VLAN 10 enter:

```
-> vlan 10 port-mac ingress-port 2/1-5
```

You can add multiple and ranges of ingress ports in the same command line. For example, to add ingress ports 2/1 through 2/5, 3/2, 5/1, and 7/4 to high availability VLAN 10 you would enter:

```
-> vlan 10 port-mac ingress-port 2/1-5 3/2 5/1 7/4
```

Removing Ingress Ports

To remove ingress ports from a high availability VLAN you use the **no** form of the **vlan port-mac ingress-port** command by entering **vlan**, followed by the VLAN's ID number, **port-mac no ingress-port**, the slot number of the port, a slash (**/**), and the port number.

For example, to remove ingress port 3/2 from high availability VLAN 10 you would enter:

```
-> vlan 10 port-mac no ingress-port 3/2
```

You can also remove multiple ingress ports by entering **vlan**, followed by the VLAN's ID number, **port-mac no ingress-port**, the slot number of the first port, a slash (**/**), the port number of the first port, a space, the slot number of the second port, a slash, and the port number of the second port. Additional ports can also be removed by specifying their slot number, a slash, and their port number preceded by a space.

For example, to remove ingress port 3/2, 5/1, and 7/4 from high availability VLAN 10 you would enter:

```
-> vlan 10 port-mac no ingress-port 3/2 5/1 7/4
```

You can also remove a range of ingress ports by entering **vlan**, followed by the VLAN's ID number, **port-mac no ingress-port**, the slot number of the first port, a slash (**/**), the port number of the first port on the NI, a hyphen (**-**), and the last port on the NI.

For example, to remove ingress port 2/1 through 2/5 from high availability VLAN 10 enter:

```
-> vlan 10 port-mac no ingress-port 2/1-5
```

You can remove multiple and ranges of ingress ports in the same command line. For example, to remove ingress ports 2/1 through 2/5, 3/2, 5/1, and 7/4 from high availability VLAN 10 you would enter:

```
-> vlan 10 port-mac no ingress-port 2/1-5 3/2 5/1 7/4
```

Note. Removing the last ingress/egress port from an HA VLAN is not allowed. Deleting the VLAN is required when there is only one ingress/egress port left in the VLAN.

Assigning and Removing Egress Ports

The following subsections describe how to assign to and remove egress ports from a high availability VLAN with the **vlan port-mac egress-port** command.

Note. Using the **vlan port-mac egress-port** command will change a standard VLAN to a high availability VLAN. Standard VLAN commands do not apply to high availability VLANs.

Assigning Egress Ports

To assign egress ports to a high availability VLAN you use the **vlan port-mac egress-port** command by entering **vlan**, followed by the VLAN's ID number, **port-mac egress-port**, the slot number of the port, a slash (/), and the port number.

For example, to add egress port 1/5 to high availability VLAN 20 you would enter:

```
-> vlan 20 port-mac egress-port 1/5
```

You can also add multiple egress ports by entering **vlan**, followed by the VLAN's ID number, **port-mac egress-port**, the slot number of the first port, a slash (/), the port number of the first port, a space, the slot number of the second port, a slash, and the port number of the second port. Additional ports can also be added by specifying their slot number, a slash, and their port number preceded by a space.

Note. Egress ports that belong to the same high availability VLAN do not need to be sequential and can be on different second-generation NI modules.

For example, to add egress port 1/5, 6/2, and 8/3 to high availability VLAN 20 you would enter:

```
-> vlan 20 port-mac egress-port 1/5 6/2 8/3
```

You can also add a range of egress ports by entering **vlan**, followed by the VLAN's ID number, **port-mac egress-port**, the slot number of the first port, a slash (/), the port number of the first port on the NI, a hyphen (**-**), and the last port on the NI.

For example, to add egress port 3/4 through 3/8 to high availability VLAN 20 enter:

```
-> vlan 20 port-mac egress-port 3/4-8
```

You can add multiple and ranges of egress ports in the same command line. For example, to add egress ports 1/5, 3/4 through 3/8, 6/2, and 8/3 to high availability VLAN 20 you would enter:

```
-> vlan 20 port-mac egress-port 1/5 3/4-8 6/2 8/3
```

Removing Egress Ports

To remove egress ports from a high availability VLAN, use the **no** form of the **vlan port-mac egress-port** command by entering **vlan**, followed by the VLAN's ID number, **port-mac no egress-port**, the slot number of the port, a slash (/), and the port number.

For example, to remove egress port 1/5 from high availability VLAN 20 you would enter:

```
-> vlan 20 port-mac no egress-port 1/5
```

You can also remove multiple egress ports by entering **vlan**, followed by the VLAN's ID number, **port-mac no egress-port**, the slot number of the first port, a slash (/), the port number of the first port, a space, the slot number of the second port, a slash, and the port number of the second port. Additional ports can also be removed by specifying their slot number, a slash, and their port number preceded by a space.

For example, to remove egress port 1/5, 6/2, and 8/3 from high availability VLAN 20 you would enter:

```
-> vlan 20 port-mac no egress-port 1/5 6/2 8/3
```

You can also remove a range of egress ports by entering **vlan**, followed by the VLAN's ID number, **port-mac no egress-port**, the slot number of the first port, a slash (/), the port number of the first port on the NI, a hyphen (**-**), and the last port on the NI.

For example, to remove egress port 3/4 through 3/8 from high availability VLAN 20 enter:

```
-> vlan 20 port-mac no egress-port 3/4-8
```

You can remove multiple and ranges of egress ports in the same command line. For example, to remove egress ports 1/5, 3/4 through 3/8, 6/2, and 8/3 from high availability VLAN 20 you would enter:

```
-> vlan 20 port-mac no egress-port 1/5 3/4-8 6/2 8/3
```

Note. Removing the last ingress/egress port from an HA VLAN is not allowed. Deleting the VLAN is required when there is only one ingress/egress port left in the VLAN.

Assigning and Removing MAC Addresses

The following subsections describe how to assign and remove MAC addresses from a high availability VLAN with the **mac-address-table port-mac vlan mac** command. Traffic that is received on ingress ports that contains a destination MAC address that matches the high availability VLAN address is sent out all egress ports that belong to the high availability VLAN.

Note. Using the **mac-address-table port-mac vlan mac** command will change a standard VLAN to a high availability VLAN. Standard VLAN commands do not apply to high availability VLANs.

Assigning MAC Addresses

To assign a MAC address to a high availability VLAN, use the **mac-address-table port-mac vlan mac** command by entering **mac-address-table port-mac vlan**, followed by the VLAN's ID number, **mac**, and the MAC address. Note that both unicast and multicast addresses are supported.

For example, to assign the MAC address 00:25:9a:5c:2f:10 to high availability VLAN 20 you would enter:

```
-> mac-address-table port-mac vlan 20 mac 00:25:9a:5c:2f:10
```

To add more than one MAC address to a high availability VLAN, enter each address on the same command line separated by a space. For example, to assign MAC addresses 00:25:9a:5c:2f:11, 00:25:9a:5c:12, and 01:00:00:3f:4c:10, to high availability VLAN 30, you would enter:

```
-> mac-address-table port-mac vlan 30 mac 00:25:9a:5c:2f:11 00:25:9a:5c:12
01:00:00:3f:4c:10.
```

Removing MAC Addresses

To remove a MAC address associated with a high availability VLAN, use the **no** form of the **mac-address-table port-mac vlan mac** command. For example, the following command removes MAC address 00:25:9a:5c:2f:10 from VLAN 20:

```
-> mac-address-table port-mac vlan 20 no mac 00:25:9a:5c:2f:10
```

To remove more than one MAC address from a high availability VLAN using a single command, enter each address on the same command line separated by a space. For example, to remove MAC addresses 00:25:9a:5c:2f:11, 00:25:9a:5c:12, and 01:00:00:3f:4c:10, from high availability VLAN 30, you would enter:

```
-> mac-address-table port-mac vlan 30 no mac 00:25:9a:5c:2f:11 00:25:9a:5c:12
01:00:00:3f:4c:10.
```

Note. Removing the last MAC address from an HA VLAN is not allowed. Deleting the VLAN is required when there is only one MAC address left.

Configuring Inter-switch Ports for HA VLANs

One scenario using HA VLANs involves configuring ingress ports on one switch and egress ports on a second switch. In this case, the ports that connect the two switches require an 802.1Q tag for the HA VLAN.

The **vlan 802.1q** command is used to tag a port with an HA VLAN. Once this is done, the tagged ports are identified as inter-switch ports and will carry HA VLAN traffic between the two switches. See “Application Example 2: Inter-Switch HA VLANs” on page 28-17 for an example of using inter-switch ports.

Note the following regarding inter-switch ports:

- Fixed ports, 802.1Q tagged ports, and link aggregates on a second-generation module are eligible to become inter-switch ports for HA VLANs.
- Once a link aggregate is tagged with an HA VLAN, it is not possible to add any more member ports to the aggregate.
- The HA VLAN and the default VLAN for an inter-switch port should participate in the same Spanning Tree instance, especially when there are redundant inter-switch connections. If this is not possible—such as when the 1x1 mode is active and there is one instance per VLAN—then ensure that the Spanning Tree state of the inter-switch port remains the same for both VLANs.

For more information about 802.1Q tagging, see Chapter 10, “Configuring 802.1Q.” For more information about Spanning Tree modes and instances, see Chapter 6, “Configuring Spanning Tree Parameters,” and Chapter 3, “Using 802.1s Multiple Spanning Tree.”

Configuring the Flood Queue Bandwidth

When the first HA VLAN is created on the switch, an ingress flood queue is automatically created for HA VLAN traffic. By default, the bandwidth size of this queue is set to 15mbps. To change the bandwidth size, use the **vlan port-mac bandwidth** command. For example, the following command sets the bandwidth value for HA VLAN 200 to 100mbps:

```
-> vlan 200 port-mac bandwidth 100
```

Note that when removing HA VLANs from the switch configuration, the flood queue remains in existence until the last HA VLAN is removed.

Application Example 1: Firewall Cluster

This section describes how to configure the traditional firewall implementation, which uses a third-party high availability firewall cluster, described in "Traditional Firewall Implementation" on page 28-7. As shown in the figure on page 28-7, traffic from the Internet comes into the switch through high availability VLAN 10 ingress ports. This VLAN has three egress ports (2/9, 2/10, and 3/5) that connect to the third-party high availability firewall cluster. The firewall cluster is connected to three ports (4/1, 5/3, 7/6) that belong to standard VLAN 20. This VLAN connects to devices within a private network.

Follow the steps below to configure the necessary high availability VLAN on an OmniSwitch.

1 Create a default VLAN for HA VLAN 10 ports with the **vlan** command as shown below:

```
-> vlan 5
```

2 Assign ports to the new default VLAN with the **vlan port default** command as shown below:

```
-> vlan 5 port default 1/1 2/9 2/10 3/5
```

3 Configure VLAN 10, which will have the ingress ports, with the **vlan** command as shown below:

```
-> vlan 10
```

4 Assign the ingress port 1/1 to VLAN 10 with the **vlan port-mac ingress-port** command as shown below:

```
-> vlan 10 port-mac ingress-port 1/1
```

5 Assign the egress ports 2/9, 2/10, and 3/5 to VLAN 10 with the **vlan port-mac egress-port** command as shown below:

```
-> vlan 10 port-mac egress-port 2/9-10 3/5
```

6 Configure standard VLAN 20, which will carry authorized traffic to the private network, with the **vlan** command as shown below:

```
-> vlan 20
```

7 Assign destination MAC addresses to VLAN 10 with the **mac-address-table port-mac vlan mac** command as shown below:

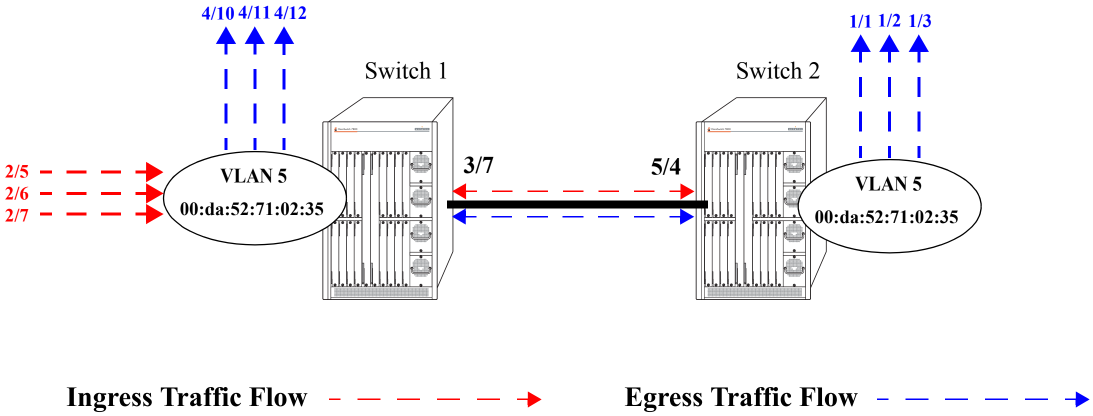
```
-> mac-address-table port-mac vlan 10 mac 00:95:2A:01:3C:10
```

Application Example 2: Inter-Switch HA VLANs

This section describes how to implement an HA VLAN configuration across two switches. As shown in the figure below:

- Ports 3/7 and 5/4 connect Switch 1 and Switch 2. Because these ports are both tagged with HA VLAN 5 (not shown), they function as inter-switch ports for VLAN 5.
- Traffic from VLAN 5 will flow in both directions through the inter-switch link provided by the 3/4 and 5/7 connection.
- VLAN 5 has three ingress ports (2/5, 2/6, and 2/7) and three egress ports (4/10, 4/11, and 4/12) on Switch 1 and three egress ports (1/1, 1/2, and 1/3) on Switch 2. In addition, VLAN 5 is also configured with an HA VLAN destination MAC address (00:da:53:71:02:35) on both switches.
- The HA VLAN ingress flood queue bandwidth size is limited to 100 Mbps.
- Traffic destined for 00:da:52:71:02:35 that is received on VLAN 5 ingress ports is forwarded to the VLAN 5 egress ports on Switch 1 *and* across the inter-switch link to VLAN 5 egress ports on Switch 2.

Note that traffic received on any ingress ports that is not destined for the HA VLAN MAC is forwarded according to switching rules.



HA VLAN Inter-Switch Configuration

Follow the steps below to configure this example inter-switch HA VLAN implementation:

1 Create a default VLAN for HA VLAN 5 ports on both Switch 1 and Switch 2 with the **vlan** command as shown below:

```
-> vlan 2
```

2 Assign ports to the new default VLAN on Switch 1 with the **vlan port default** command as shown below:

```
-> vlan 2 port default 2/5-7 3/7 4/10-12
```

3 Assign ports to a new default VLAN on Switch 2 with the **vlan port default** command as shown below:

```
-> vlan 2 port default 1/1-3 5/4
```

4 Configure VLAN 5, which will become an HA VLAN, on both Switch 1 and Switch 2 with the **vlan** command as shown below:

```
-> vlan 5
```

5 Assign the ingress ports 2/5, 2/6, and 2/7 to VLAN 5 on Switch 1 with the **vlan port-mac ingress-port** command as shown below:

```
-> vlan 5 port-mac ingress-port 2/5-7
```

6 Assign the egress ports 4/10, 4/11, and 4/12 to VLAN 5 on Switch 1 with the **vlan port-mac egress-port** command as shown below:

```
-> vlan 5 port-mac egress-port 4/10-12
```

7 Tag port 3/7 on Switch 1 with VLAN 5 using the **vlan 802.1q** command as shown below:

```
-> vlan 5 802.1q 3/7
```

8 Assign the egress ports 1/1, 1/2, and 1/3 to VLAN 5 on Switch 2 with the **vlan port-mac egress-port** command as shown below:

```
-> vlan 5 port-mac egress-port 1/1-3
```

9 Tag port 5/4 on Switch 2 with VLAN 5 using the **vlan 802.1q** command as shown below:

```
-> vlan 5 802.1q 5/4
```

10 Assign the HA VLAN destination MAC address to VLAN 5 on both Switch 1 and Switch 2 using the **mac-address-table port-mac vlan mac** command as shown below:

```
-> mac-address-table port-mac vlan 10 mac 00:da:52:71:02:35
```

11 Set the HA VLAN ingress flood queue bandwidth size to 100 Mbps using the **vlan port-mac bandwidth** command as shown below:

```
-> vlan 5 port-mac bandwidth 100
```

Displaying High Availability VLAN Status and Statistics

You can use CLI **show** commands to display the current configuration and statistics of high availability VLANs on a switch. These commands include the following:

show mac-address-table port-mac	Displays the status and configuration of high availability VLANs.
show vlan	Displays a list of all VLANs configured on the switch and the status of related VLAN properties (e.g., admin and Spanning Tree status and router port definitions).
show vlan port	Displays a list of VLAN port assignments.

To display the status and configuration of high availability VLANs you use the **show mac-address-table port-mac** command. To display the status and configuration of all high availability VLANs on a switch enter:

```
-> show mac-address-table port-mac
```

A screen similar to the following will be displayed:

```
Port mac configuration for vlan 10

Bandwidth : 15 MB/sec

   Ingress Port list:
           3/5  3/7
   Egress Port list:
           3/9  3/6
   Mac Address list:
           00:DA:95:3C:44:55
           00:13:14:34:5E:78
           01:23:45:C1:17:21

Port mac configuration for vlan 20

Bandwidth : 15 MB/sec

   Ingress Port list:
           1/4  8/2
   Egress Port list:
           3/9  3/6
   Mac Address list:
           00:11:22:33:44:55
           07:23:14:34:31:25
           00:23:45:67:43:04
```

To display the status and configuration of a single high availability VLAN enter **show mac-address-table port-mac vlan** followed by the VLAN’s ID number. For example, to display the status and configuration of high availability VLAN 10 enter

```
-> show mac-address-table port-mac vlan 10
```

A screen similar to the following will be displayed:

```
Port mac configuration for vlan 10

Bandwidth : 15 MB/sec

   Ingress Port list:
           3/5  3/7
   Egress Port list:
           3/9  3/6
   Mac Address list:
           00:11:22:33:44:55
           00:13:14:34:34:78
           01:23:45:67:11:21
```

Note. See the *OmniSwitch CLI Reference Guide*.for complete syntax for the **show mac-address-table port-mac** command.

29 Diagnosing Switch Problems

Several tools are available for diagnosing problems that may occur with the switch. These tools include

- Port Mirroring
- Remote Monitoring (RMON) probes
- Switch Health Monitoring

Port mirroring copies all incoming and outgoing traffic from a single mirrored Ethernet port to a second mirroring Ethernet port, where it can be monitored with a Remote Network Monitoring (RMON) probe or network analysis device without disrupting traffic flow on the mirrored port. Switch Health monitoring software checks previously-configured threshold levels for the switch's consumable resources, and notifies the Network Monitoring Station (NMS) if those limits are violated.

In This Chapter

This chapter describes the port mirroring, remote monitoring (RMON) probes, and switch health features and how to configure them through the Command Line Interface (CLI).

Configuration procedures described in this chapter include:

- Creating or Deleting a Port Mirroring Session—see "Creating a Mirroring Session" on page 29-12 or "Deleting a Mirroring Session" on page 29-15.
- Protection from Spanning Tree changes (Port Mirroring)—see "Unblocking Ports (Protection from Spanning Tree)" on page 29-12.
- Enabling or Disabling Port Mirroring Status—see "Enabling or Disabling Mirroring Status" on page 29-13 or "Disabling a Mirroring Session (Disabling Mirroring Status)" on page 29-13.
- Configuring Port Mirroring Direction—see "Configuring Port Mirroring Direction" on page 29-14.
- Enabling or Disabling a Port Mirroring Session—see "Enabling or Disabling a Port Mirroring Session (Shorthand)" on page 29-14.

- Enabling or Disabling RMON Probes—see "Enabling or Disabling RMON Probes" on page 29-18.
- Configuring Resource Threshold Limits (Switch Health)—see "Configuring Resource and Temperature Thresholds" on page 29-25.
- Configuring Sampling Intervals—see "Configuring Sampling Intervals" on page 29-27.
- Resetting Health Statistics—see "Resetting Health Statistics for the Switch" on page 29-29.

For information about additional Diagnostics features such as Switch Logging and System Debugging/ Memory Management commands, see Chapter 30, "Using Switch Logging" and Chapter 31, "Monitoring Memory."

Port Mirroring Overview

The following sections detail the specifications, defaults, and quick set up steps for the port mirroring feature. Detailed procedures are found in "Port Mirroring" on page 29-9.

Note. A port that is part of an aggregate link cannot be mirrored.

Port Mirroring Specifications

Ports Supported	Ethernet (10 Mbps)/Fast Ethernet (100 Mbps)/ Gigabit Ethernet (1 Gb/1000 Mbps)
Mirroring Sessions supported	Up to 4 unidirectional sessions *or* 2 bidirectional sessions.
Port Capacity Requirements	Mirrored *(monitored)* and mirroring *(monitoring)* ports must be of identical capacity (both ports support identical Mbps rates) or the Mirroring port must be of higher capacity than the mirrored port. (Example: A mirrored Fast Ethernet port supports 100 Mbps, while a Mirroring Gigabit Ethernet port supports 1000 Mbps).
Range of Unblocked VLAN IDs	1 to 4094.

Port Mirroring Defaults

The following table shows port mirroring default values.

Global Port Mirroring Defaults

Parameter Description	CLI Command	Default Value/Comments
Mirroring Session Creation	**port mirroring source destination**	No Mirroring Sessions Configured
Protection from Spanning Tree (Spanning Tree Disable)	**port mirroring source destination**	Spanning Tree Enabled
Mirroring Status Configuration	**port mirroring source destination**	Disabled
Mirroring Session Configuration	**port mirroring**	Disabled
Mirroring Session Deletion	**port mirroring**	No Mirroring Sessions Configured

Quick Steps for Configuring Port Mirroring

1 Create a port mirroring session. Be sure to specify the port mirroring session ID, source *(mirrored)* and destination *(mirroring)* slot/ports, and unblocked VLAN ID (*optional*—protects the mirroring session from changes in Spanning Tree if the mirroring port will monitor mirrored traffic on an RMON probe belonging to a different VLAN). For example:

```
-> port mirroring 6 source 2/3 destination 6/4 unblocked 7
```

2 Enable the port mirroring session.

```
-> port mirroring 6 enable
```

Note. *Optional.* To verify the port mirroring configuration, enter **show port mirroring status** followed by the port mirroring session ID number. The display is similar to the one shown below:

```
Session   Mirrored   Mirroring      Mirror     Mirroring  Mirroring
          slot/port  slot/port     Direction      Vlan      Status
----------+----------+----------+--------------+----------+----------
6.         2/3       6/4            bidirectional 7         ON
```

For more information about this command, see “Displaying Port Mirroring Status” on page 29-15 or the “Port Mirroring and Monitoring Commands” chapter in the *OmniSwitch CLI Reference Guide.*

Remote Monitoring (RMON) Overview

The following sections detail the specifications, defaults, and quick set up steps for the RMON feature. Detailed procedures are found in "Remote Monitoring (RMON)" on page 29-16.

RMON Specifications

RFCs Supported	2819 - Remote Network Monitoring Management Information Base
RMON Functionality Supported	Basic RMON 4 group implementation –Ethernet Statistics group –History (Control and Statistics) group –Alarms group –Events group
RMON Functionality Not Supported	RMON 10 group* RMON2* –Host group –HostTopN group –Matrix group –Filter group –Packet Capture group (*An external RMON probe that includes RMON 10 group and RMON2 may be used where full RMON probe functionality is required.)
Flavor (Probe Type)	Ethernet/History/Alarm
Status	Active/Creating/Inactive
History Control Interval (seconds)	1 to 3600
History Sample Index Range	1 to 65535
Alarm Interval (seconds)	1 to 2147483647
Alarm Startup Alarm	Rising Alarm/Falling Alarm/ RisingOrFalling Alarm
Alarm Sample Type	Delta Value/Absolute
RMON Traps Supported	RisingAlarm/FallingAlarm These traps are generated whenever an Alarm entry crosses either its Rising Threshold or its Falling Threshold and generates an event configured for sending SNMP traps.

RMON Probe Defaults

The following table shows Remote Network Monitoring default values.

Global RMON Probe Defaults

Parameter Description	CLI Command	Default Value/Comments
RMON Probe Configuration	**rmon probes**	No RMON probes configured.

Quick Steps for Enabling/Disabling RMON Probes

1 Enable an inactive (or disable an active) RMON probe, where necessary. You can also enable or disable all probes of a particular flavor, if desired. For example:

```
-> rmon probes stats 4005 enable

-> rmon probes history disable
```

2 To verify the RMON probe configuration, enter the **show rmon probes** command with the keyword for the type of probe. For example, to display the statistics probes, enter the following:

```
-> show rmon probes stats
```

The display is similar to the one shown below:

```
Entry  Slot/Port   Flavor      Status     Duration        System Resources
-----+-----------+-----------+----------+---------------+---------------------
4001   4/1         Ethernet    Active     00:25:00        275 bytes
4008   4/8         Ethernet    Active     00:25:00        275 bytes
4005   4/5         Ethernet    Active     00:03:03        275 bytes
```

3 To view statistics for a particular RMON probe, enter the **show rmon probes** command, with the keyword for the type of probe, followed by the entry number for the desired RMON probe. For example:

```
-> show rmon probes 4005
```

4 The display will appear similar to the one shown below:

```
Probe’s Owner: Falcon Switch Auto Probe on Slot 4, Port 5
   Entry 4005
      Flavor = Ethernet, Status = Active
      Time = 0 hrs 03 mins,
      System Resources (bytes) = 275
```

For more information about these commands, see “Displaying a List of RMON Probes” on page 29-19, “Displaying Statistics for a Particular RMON Probe” on page 29-20 or the “RMON Commands” chapter in the *OmniSwitch CLI Reference Guide.*

Switch Health Overview

The following sections detail the specifications, defaults, and quick set up steps for the switch health feature. Detailed procedures are found in "Monitoring Switch Health" on page 29-23.

Switch Health Specifications

Health Functionality Supported	–Switch level CPU Utilization Statistics (percentage);
	–Switch/module/port level Input Utilization Statistics (percentage);
	–Switch/module/port level Input/Output Utilization Statistics (percentage);
	–Switch level Memory Utilization Statistics (percentage);
	–Device level (e.g., Chassis/CMM) Temperature Statistics (Celsius).

Monitored Resource Utilization Levels	–Most recent utilization level; –Average utilization level during last minute; –Average utilization level during last hour; –Maximum utilization level during last hour.
Resource Utilization Raw Sample Values	Saved for previous 60 seconds.
Resource Utilization Current Sample Values	Stored.
Resource Utilization Maximum Utilization Value	Calculated for previous 60 seconds and stored.
Utilization Value = 0	Indicates that none of the resource was measured for the period.
Utilization Value = 1	Indicates that a non-zero amount of the resource (less than 2%) was measured for the period.
Percentage Utilization Values	Calculated based on Resource Measured During Period/Total Capacity.
Resource Threshold Levels	Apply automatically across all levels of switch (switch/module/port).
Rising Threshold Crossing	A Resource Threshold was exceeded by its corresponding utilization value in the current cycle.
Falling Threshold Crossing	A Resource Threshold was exceeded by its corresponding utilization value in the previous cycle, but is not exceeded in the current cycle.
Threshold Crossing Traps Supported	Device, module, port-level threshold crossings.

Switch Health Defaults

The following table shows Switch Health default values.

Global Switch Health Defaults

Parameter Description	CLI Command	Default Value/Comments
Resource Threshold Limit Configuration	**health threshold**	80 percent
Sampling Interval Configuration	**health interval**	5 seconds
Switch Temperature	**health interval**	50 degrees Celsius

Quick Steps for Configuring Switch Health

1 Display the health threshold limits, health sampling interval settings and/or health statistics for the switch, depending on the parameters you wish to modify. (For best results, note the default settings for future reference.) For example:

```
-> show health threshold
```

The default settings for the command you entered will be displayed. For example:

```
Rx Threshold                = 80
TxRx Threshold              = 80
Memory Threshold            = 80
CPU Threshold               = 80
Temperature Threshold       = 50
```

2 Enter the appropriate command to change the desired health threshold or health sampling interval parameter settings, or reset all health statistics for the switch. For example:

```
-> health threshold memory 85
```

Note. *Optional.* To verify the Switch Health configuration, enter **show health threshold** followed by the parameter you modified (e.g., **memory**). The display is similar to the one shown below:

```
Memory Threshold           = 85
```

For more information about this command, see "Displaying Health Threshold Limits" on page 29-26 or the "Health Monitoring Commands" chapter in the *OmniSwitch CLI Reference Guide.*

Port Mirroring

You can set up port mirroring for any pair of Ethernet ports within the same switch chassis. Ethernet ports supporting port mirroring include:

- 10BaseT, 100BaseTx, and 1000Baset using RJ-45 connectors
- 100BaseFX using MT-RJ connectors
- 1000BaseSX and 1000BaseLX using either SC connectors (GBICs) or LC connectors (MiniGBICs).

Note. Port mirroring is not supported on OS8-10GNI-UR1 modules.

When port mirroring is enabled, the active "mirrored" port transmits and receives network traffic normally, and the "mirroring" port receives a copy of all transmit and receive traffic to the active port. You can connect an RMON probe or network analysis device to the mirroring port to see an exact duplication of traffic on the mirrored port without disrupting network traffic to and from the mirrored port.

Note. A port that is part of an aggregate link cannot be mirrored.

Port mirroring runs in software on the Chassis Management Module (CMM) and on all Network Interface (NI) modules on the OmniSwitch, and is supported for Ethernet (10 Mbps), Fast Ethernet (100 Mbps) and Gigabit Ethernet (1000 Mpbs) ports. Up to 4 unidirectional sessions *or* 2 bidirectional sessions are supported.

When a port mirroring session is configured, both the mirrored and mirroring ports are automatically included in the same VLAN. Both ports must be identical in capacity (e.g., both ports support 10Mbps, 100Mbps or 1000Mbps) *or* the mirroring port must support a higher capacity than the mirrored port (e.g., a mirrored Fast Ethernet port supports 100Mbps and a mirroring Gigabit Ethernet port supports 1000Mbps).

Note. Both the mirrored and mirroring ports must be connected and up (enabled) to start mirroring.

Port Mirroring and 802.1Q Packets

Port mirroring of 802.1Q packets can produce unintended results on OmniSwitch 7700/7800/8800 switches. Indeed if a packet is untagged and arrives on a tagged port, the Hardware Routing Engine (HRE) inserts that the default VLAN ID of the ingress port. When the HRE sends an unmodified packet to the port mirroring application the software will undo changes that HRE has made.

If a packet arrives on a tagged port and it is untagged, the HRE inserts the default 802.1Q tag for this port. If the the "t" bit is unset and the "q" bit bit set, the port mirroring application removes the 802.1Q bytes and set the VLAN ID.

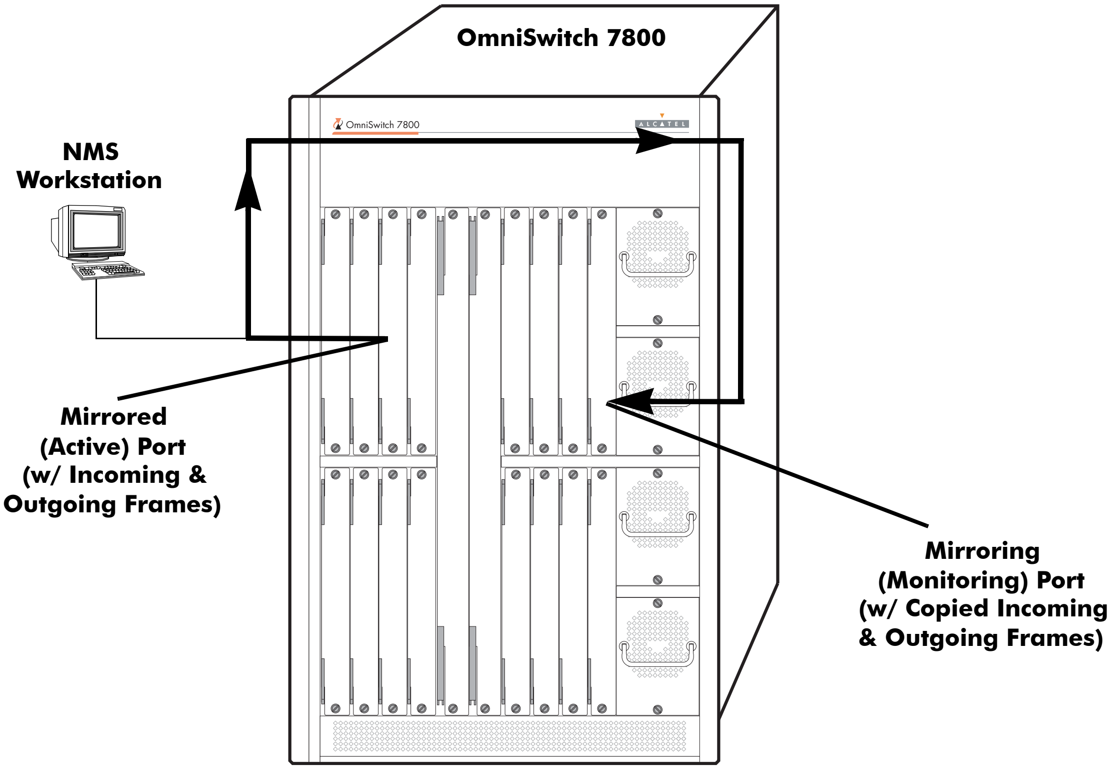
If the destination address is known then port mirroring will process the packet according to the information available in the Destination Address (DA) table entry. If the entry indicates that a Layer 3 or 4 lookup is necessary, then the software cannot correctly process the packet.

How Port Mirroring Works

When a frame is received on a mirrored port, it is copied and sent to the mirroring port. The received frame is actually transmitted twice across the switch backplane–once for normal bridging and then again to the mirroring port.

When a frame is transmitted by the mirrored port, a copy of the frame is made, tagged with the mirroring port as the destination, and sent back over the switch backplane to the mirroring port. The diagram below illustrates the data flow between the mirrored and mirroring ports.

Note that when port mirroring is enabled, there may be some performance degradation, since all frames received and transmitted by the mirrored port need to be copied and sent to the mirroring port.


OmniSwitch 7800
OmniSwitch 7800
ALCATEL
NMS Workstation
Mirrored (Active) Port (w/ Incoming & Outgoing Frames)
Mirroring (Monitoring) Port (w/ Copied Incoming & Outgoing Frames)

Relationship between Mirrored and Mirroring Ports

What Happens to the Mirroring Port

When you set up port mirroring and attach cables to the mirrored and mirroring ports, the mirroring port remains enabled and part of the Bridging Spanning Tree until you protect it from Spanning Tree updates by specifying an unblocked VLAN as part of the configuration command line. Once you've done so, the mirroring port is administratively disabled and no longer part of the Bridging Spanning Tree. The mirroring port does not transmit or receive any traffic on its own.

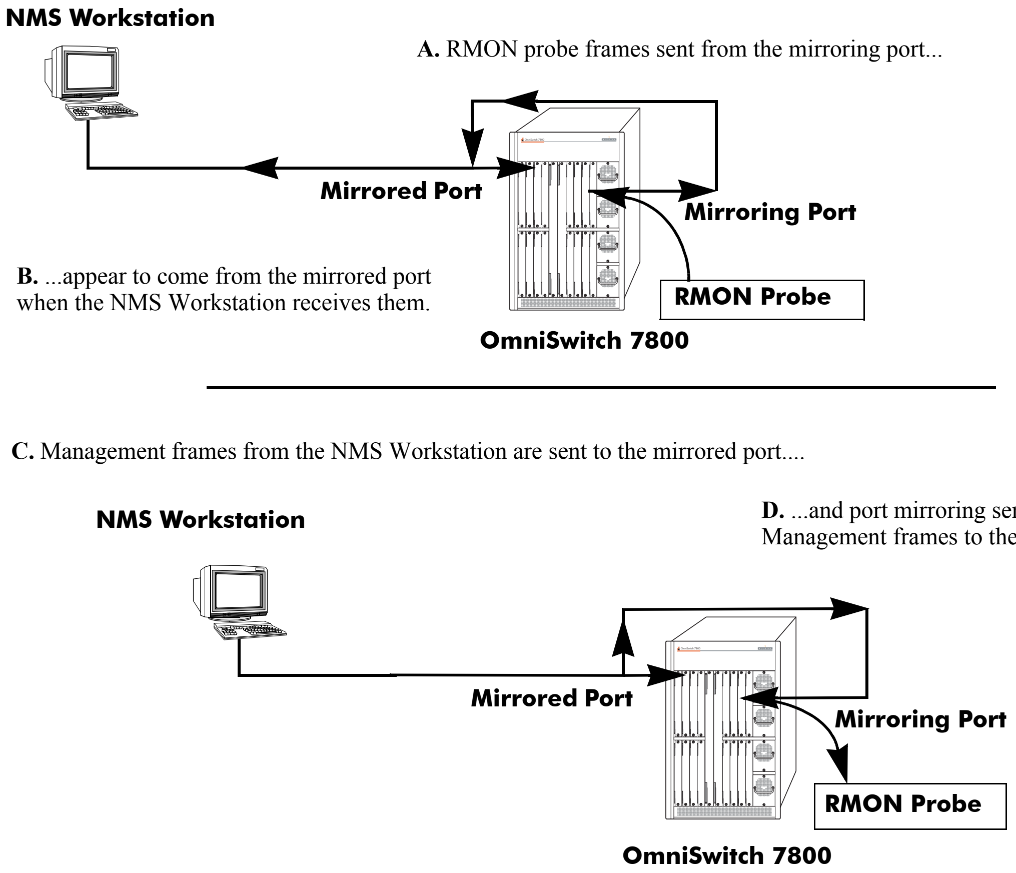
Note. If a cable is not attached to the mirrored port, port mirroring will not take place. In this case, the mirroring port reverts back to its normally operational state and will bridge frames as if port mirroring were disabled.

Using Port Mirroring with External RMON Probes

Port mirroring is a helpful monitoring tool when used in conjunction with an external RMON probe. Once you set up port mirroring, the probe can collect all relevant RMON statistics for traffic on the mirrored port. You can also move the mirrored port so that the mirroring port receives data from different ports. In this way, you can roam the switch and monitor traffic at various ports.

Note. If the mirroring port will monitor mirrored traffic on an RMON probe belonging to a different VLAN than the mirrored port, it should be protected from blocking due to Spanning Tree updates. See "Unblocking Ports (Protection from Spanning Tree)" on page 29-12 for details.

The following diagram illustrates how port mirroring can be used with an external RMON probe, to copy RMON probe frames and Management frames to and from the mirroring and mirrored ports. Frames received from an RMON probe attached to the mirroring port can be seen as being received by the mirrored port. These frames from the mirroring port are marked as if they are received on the mirrored port before being sent over the switch backplane to an NMS station. Therefore, management frames destined for the RMON probe are first forwarded out the mirrored port. After being received on the mirrored port, copies of the frames are mirrored out the mirroring port—the probe attached to the mirroring port receives the management frames.



A. RMON probe frames sent from the mirroring port...

B. ...appear to come from the mirrored port when the NMS Workstation receives them.

C. Management frames from the NMS Workstation are sent to the mirrored port....

D. ...and port mirroring sends copies of the Management frames to the mirroring port.

Port Mirroring Using External RMON Probe

Creating a Mirroring Session

Before port mirroring can be used, it is necessary to create a port mirroring session. The **port mirroring source destination** command can be used to create a mirroring session between a mirrored (active) port and a mirroring port. Up to 4 unidirectional sessions *or* 2 bidirectional sessions are supported.

Note. To prevent the mirroring (destination) port from being blocked due to Spanning Tree changes, be sure to specify the VLAN ID number (from 1 to 4094) for the port that will remain **unblocked** (protected from these changes while port mirroring is active). This parameter is optional; if it is not specified, changes resulting from Spanning Tree could cause the port to become blocked *(*default*)*. See **Unblocking Ports (Protection from Spanning Tree)** below for details.

To create a mirroring session, enter the **port mirroring source destination** command, and include the port mirroring session ID number and the source and destination slot/ports, as shown in the following example:

```
-> port mirroring 6 source 2/3 destination 6/4
```

This command line specifies mirroring session 6, with the source (mirrored) port located in slot 2/port 3, and the destination (mirroring) port located in slot 6/port 4.

Note. Neither the mirrored nor the mirroring ports can be a mobile port. See Chapter 7, “Assigning Ports to VLANs,” for information on mobile ports.

As an option, the same destination port can be shared by up to four sessions. In the following example sessions 1, 5, and 6 mirror source ports 1/2, 2/1, and 2/3 on destination port 2/4:

```
-> port mirroring 1 source 1/2 destination 2/4
-> port mirroring 5 source 2/1 destination 2/4
-> port mirroring 6 source 2/3 destination 2/4
```

Unblocking Ports (Protection from Spanning Tree)

If the mirroring port will monitor mirrored traffic on an RMON probe belonging to a different VLAN than the mirrored port, it should be protected from blocking due to Spanning Tree updates. To create a mirroring session that protects the mirroring port from being blocked *(default)* due to changes in Spanning Tree, enter the **port mirroring source destination** command, and include the port mirroring session ID number, source and destination slot/ports, and unblocked VLAN ID number, as shown in the following example:

```
-> port mirroring 6 source 2/3 destination 6/4 unblocked 750
```

This command line specifies mirroring session 6, with the source (mirrored) port located in slot 2/port 3, and the destination (mirroring) port located in slot 6/port 4. The mirroring port on VLAN 750 is protected from Spanning Tree updates.

Note. If the unblocked VLAN identifier is not specified, the mirroring port could be blocked due to changes in Spanning Tree.

Enabling or Disabling Mirroring Status

Mirroring Status is the parameter by which you can enable or disable a mirroring session (i.e., turn port mirroring on or off). There are two ways to do this:

- *Creating a Mirroring Session and Enabling Mirroring Status* or *Disabling a Mirroring Session (Disabling Mirroring Status)*. These procedures are described below and on the following page.
- *Enabling or Disabling a Port Mirroring Session*—"shorthand" versions of the above commands that require fewer keystrokes. Only the port mirroring session ID number needs to be specified, rather than the entire original command line syntax (e.g., source and destination slot/ports and optional unblocked VLAN ID number). See "Enabling or Disabling a Port Mirroring Session (Shorthand)" on page 29-14 for details.

Note. A port that is part of an aggregate link cannot be mirrored.

Creating a Mirroring Session and Enabling Mirroring Status

To create a mirroring session with protection against blocking and mirroring status enabled (turning port mirroring on), enter the **port mirroring source destination** command. Include the port mirroring session ID number, the source and destination slot/ports, optional unblocked VLAN ID number and **enable**, as shown in the following example:

```
-> port mirroring 6 source 2/3 destination 6/4 unblocked 750 enable
```

This command line specifies mirroring session 6, with the mirrored (active) port located in slot 2/port 3, and the mirroring port located in slot 6/port 4. The mirroring port on VLAN 750 is protected from Spanning Tree updates, and Mirroring Status is enabled (i.e., port mirroring is turned ON). Note that the port mirroring session identifier and slot/port locations of the designated interfaces must always be specified.

Note. Port mirroring session parameters cannot be modified when a mirroring session is enabled. Before you can modify parameters, the mirroring session must be disabled.

Disabling a Mirroring Session (Disabling Mirroring Status)

To disable the mirroring status of the configured session between a mirrored port and a mirroring port (turning port mirroring off), use the **port mirroring source destination** command. Be sure to include the port mirroring session ID number and the keyword **disable**.

In this example the command specifies port mirroring session 6, with the mirrored (active) port located in slot 2/port 3, and the mirroring port located in slot 6/port 4. The mirroring status is disabled (i.e., port mirroring is turned off):

```
-> port mirroring 6 source disable
```

Note. You can modify the parameters of a port mirroring session that has been disabled.

Keep in mind that the port mirroring session configuration remains valid, even though port mirroring has been turned off. Note that the port mirroring session identifier and slot/port locations of the designated interfaces must always be specified.

Note. Note that the port mirroring session identifier and slot/port locations of the designated interfaces must always be specified.

Configuring Port Mirroring Direction

By default, port mirroring sessions are bidirectional. To configure the direction of a port mirroring session between a mirrored port and a mirroring port use the **port mirroring source destination** command by entering port mirroring followed by the port mirroring session ID number, the source and destination slot/ports, and **bidirectional**, **inport**, or **outport**.

Note. Optionally, you can also specify the optional unblocked VLAN ID number and either **enable** or **disable** on the same command line.

In this example the command specifies port mirroring session 6, with the mirrored (active) port located in slot 2/port 3, and the mirroring port located in slot 6/port 4. The mirroring direction is unidirectional and inward bound:

```
-> port mirroring 6 source 2/3 destination 6/4 inport
```

In this example the command specifies port mirroring session 6, with the mirrored (active) port located in slot 2/port 3, and the mirroring port located in slot 6/port 4. The mirroring direction is unidirectional and outward bound:

```
-> port mirroring 6 source 2/3 destination 6/4 outport
```

You can use the bidirectional keyword to restore a mirroring session to its default bidirectional configuration. For example:

```
-> port mirroring 6 source 2/3 destination 6/4 bidirectional
```

Note. Note that the port mirroring session identifier and slot/port locations of the designated interfaces must always be specified.

Enabling or Disabling a Port Mirroring Session (Shorthand)

Once a port mirroring session configuration has been created, this command is useful for enabling or disabling it (turning port mirroring on or off) without having to re-enter the source and destination ports and unblocked VLAN ID command line parameters.

To enable a port mirroring session, enter the **port mirroring** command, followed by the port mirroring session ID number and the keyword **enable**. The following command enables port mirroring session 6 (turning port mirroring on):

```
-> port mirroring 6 enable
```

Note. Port mirroring session parameters cannot be modified when a mirroring session is enabled. Before you can modify parameters, the mirroring session must be disabled.

To disable a port mirroring session, enter the **port mirroring** command, followed by the port mirroring session ID number and the keyword **disable**. The following command disables port mirroring session 6 (turning port mirroring off):

```
   -> port mirroring 6 disable
```

Displaying Port Mirroring Status

To display port mirroring status, use the **show port mirroring status** command. To display all port mirroring sessions enter:

```
   -> show port mirroring status
```

A screen similar to the following will be displayed:

```
   Session   Mirrored   Mirroring      Mirror     Mirroring  Mirroring
             slot/port  slot/port     Direction     Vlan       Status
   ----------+----------+----------+--------------+----------+----------
   6.        1/23       1/24          bidirectional NONE       OFF
   9.        2/1         2/11         inport         7         ON
```

To display a specific session enter **show port mirroring status** followed by the port mirroring session ID number. For example:

```
   -> show port mirroring status 6

   Session   Mirrored   Mirroring      Mirror     Mirroring  Mirroring
             slot/port  slot/port     Direction     Vlan       Status
   ----------+----------+----------+--------------+----------+----------
   6.        1/23       1/24          bidirectional NONE       OFF
```

In this example, the status of the mirrored and mirroring ports in mirroring session 6 is displayed. The locations of the mirrored and mirroring ports are shown, (slot 1, port 14 and slot 1, port 16, respectively), along with the mirroring VLAN ID number (5), direction, and mirroring Status (port mirroring is OFF).

Deleting a Mirroring Session

The **no** form of the **port mirroring** command can be used to delete a previously created mirroring session configuration between a mirrored port and a mirroring port.

To delete a mirroring session, enter the **no port mirroring** command, followed by the port mirroring session ID number. For example:

```
   -> no port mirroring 6
```

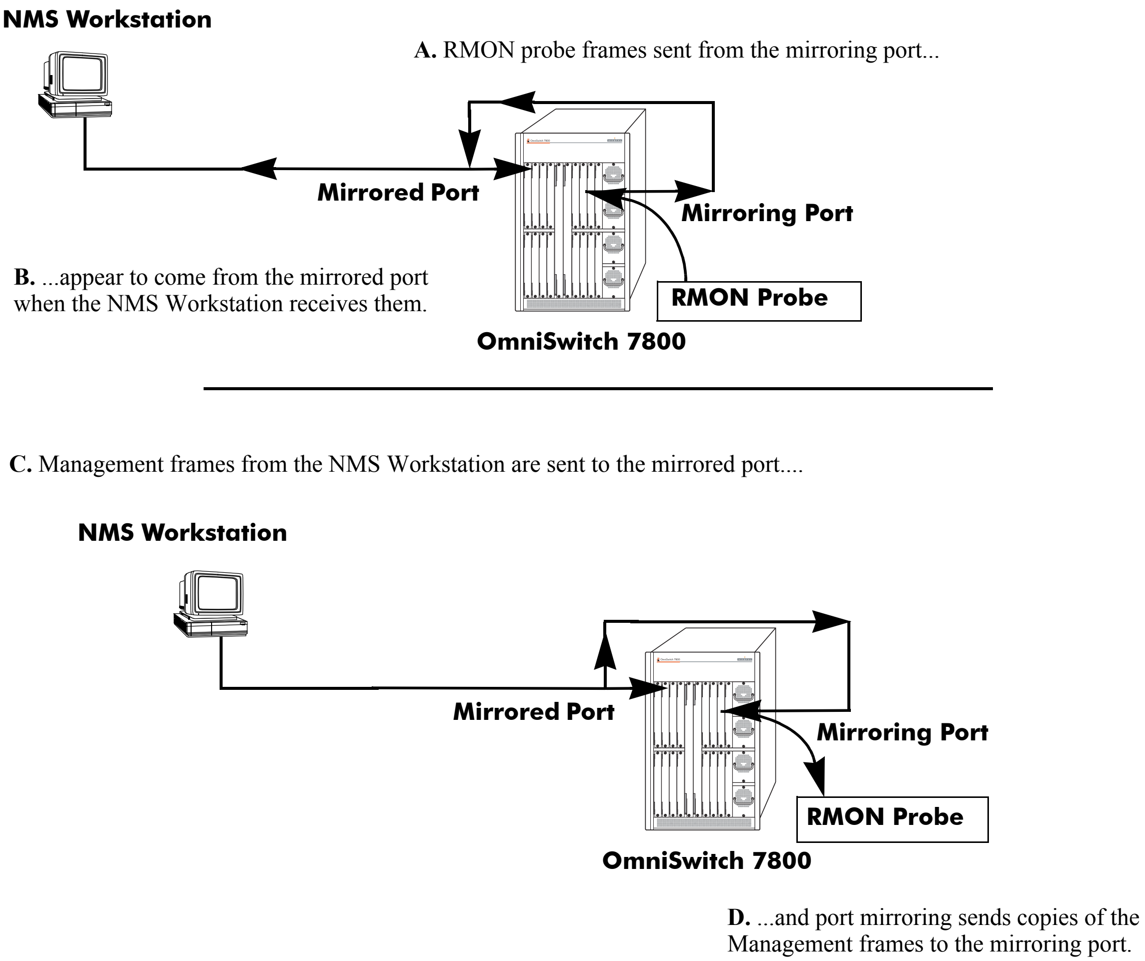
In this example, port mirroring session 6 is deleted.

Note. The port mirroring session identifier must always be specified.

Remote Monitoring (RMON)

Remote Network Monitoring (RMON) is an SNMP protocol used to manage networks remotely. *RMON probes* can be used to collect, interpret and forward statistical data about network traffic from designated active ports in a LAN segment to an NMS (Network Management System) application for monitoring and analysis without negatively impacting network performance. RMON software is fully integrated in the Chassis Management Module (CMM) and works with the Ethernet software to acquire statistical information. However, it does not monitor the CMM module's onboard 10/100BaseT Ethernet Management Port (which is reserved for management purposes).

The following diagram illustrates how an External RMON Probe can be used with port mirroring to copy RMON probe frames and Management frames to and from the mirroring and mirrored ports. Frames received from an RMON probe attached to the mirroring port can be seen as being received by the mirrored port. These frames from the mirroring port are marked as if they are received on the mirrored port before being sent over the switch backplane to an NMS station. Therefore, management frames that are destined for the RMON probe are first forwarded out the mirrored port. After being received on the mirrored port, copies of the frames are mirrored out the mirroring port—the probe attached to the mirroring port receives the management frames.


NMS Workstation
A. RMON probe frames sent from the mirroring port...
Mirrored Port
Mirroring Port
B. ...appear to come from the mirrored port when the NMS Workstation receives them.
RMON Probe
OmniSwitch 7800
C. Management frames from the NMS Workstation are sent to the mirrored port....
NMS Workstation
Mirrored Port
Mirroring Port
RMON Probe
OmniSwitch 7800
D. ...and port mirroring sends copies of the Management frames to the mirroring port.

Port Mirroring Using External RMON Probe

RMON probes can be enabled or disabled via commands. Configuration of Alarm threshold values for RMON traps is a function reserved for RMON-monitoring NMS stations.

This feature supports basic RMON 4 group implementation in compliance with RFC 2819, including the **Ethernet Statistics**, **History** (Control & Statistics), **Alarms** and **Events** groups *(described below)*.

Note. RMON 10 group and RMON2 are not implemented in the current release. An external RMON probe that includes RMON 10 group and RMON2 may be used where full RMON probe functionality is required.

Ethernet Statistics

Ethernet statistics probes are created whenever new ports are inserted and activated in the chassis. When a port is removed from the chassis or deactivated, the Ethernet statistics group entry associated with the physical port is invalidated and the probe is deleted.

The Ethernet statistics group includes port utilization and error statistics measured by the RMON probe for each monitored Ethernet interface on the switch. Examples of these statistics include CRC (Cyclic Redundancy Check)/alignment, undersized/oversized packets, fragments, broadcast/multicast/unicast, and bandwidth utilization statistics.

History (Control & Statistics)

The History (Control & Statistics) group controls and stores periodic statistical samplings of data from various types of networks. Examples include Utilization, Error Count and Frame Count statistics.

Alarm

The Alarm group collects periodic statistical samples from variables in the probe and compares them to previously configured thresholds. If a sample crosses a previously configured threshold value, an Event is generated. Examples include Absolute or Relative Values, Rising or Falling Thresholds on the Utilization Frame Count and CRC Errors.

Event

The Event group controls generation and notification of events from the switch to NMS stations. For example, customized reports based on the type of Alarm can be generated, printed and/or logged.

Note. The following RMON groups are not implemented: **Host**, **HostTopN**, **Matrix**, **Filter** and **Packet Capture**.

Enabling or Disabling RMON Probes

To enable or disable an individual RMON probe, enter the **rmon probes** command. Be sure to specify the type of probe (**stats**/**history**/**alarm**), followed by the entry number (optional), as shown in the following examples.

The following command enables RMON Ethernet Statistics probe number 4012:

```
-> rmon probes stats 4012 enable
```

The following command disables RMON History probe number 10240:

```
-> rmon probes history 10240 disable
```

The following command enables RMON Alarm probe number 11235:

```
-> rmon probes alarm 11235 enable
```

To enable or disable an entire group of RMON probes of a particular flavor type (such as Ethernet Statistics, History or Alarm), enter the command **without** specifying an *entry-number*, as shown in the following examples.

The following command disables all currently defined (disabled) RMON Ethernet Statistics probes:

```
-> rmon probes stats disable
```

The following command enables all currently defined (disabled) RMON History probes:

```
-> rmon probes history enable
```

The following command enables all currently defined (disabled) RMON Alarm probes:

```
-> rmon probes alarm enable
```

Notes. Network activity on subnetworks attached to an RMON probe can be monitored by Network Management Software (NMS) applications. RMON will not monitor activities on the CMM onboard Ethernet Management Port.

Displaying RMON Tables

Two separate commands can be used to retrieve and view Remote Monitoring data: **show rmon probes** and **show rmon events**. The retrieved statistics appear in a *table* format (a collection of related data that meets the criteria specified in the command you entered). These RMON tables can display the following kinds of data (depending on the criteria you've specified):

- The **show rmon probes** command can display a list of current RMON probes, or statistics for a particular RMON probe.
- The **show rmon events** command can display a list of RMON events (actions that occur in response to Alarm conditions detected by an RMON probe), or statistics for a particular RMON event.

Displaying a List of RMON Probes

To view a list of current RMON probes, enter the **show rmon probes** command with the probe type without specifying an entry number for a particular probe.

For example, to show a list of the statistics probes, enter:

```
-> show rmon probes stats
```

A display showing all current statistics RMON probes should appear, as shown in the following example:

```
Entry   Slot/Port    Flavor      Status      Duration       System Resources

-------+-----------+-----------+----------+---------------+---------------------
4001    4/1          Ethernet    Active      00:25:00       275 bytes
4008    4/8          Ethernet    Active      00:25:00       275 bytes
4005    4/5          Ethernet    Active      00:25:00       275 bytes
```

This table entry displays probe statistics for all probes on the switch. The probes are active, utilize 275 bytes of memory, and 25 minutes have elapsed since the last change in status occurred.

To show a list of the history probes, enter:

```
-> show rmon probes history
```

A display showing all current history RMON probes should appear, as shown in the following example:

```
Entry   Slot/Port    Flavor      Status      Duration       System Resources

-------+-----------+-----------+----------+---------------+---------------------
4001    4/1          History     Active      00:25:00       601 bytes
4008    4/8          History     Active      00:25:00       601 bytes
4005    4/5          History     Active      00:25:00       601 bytes
```

The second table entry displays statistics for RMON History probe on the switch. This probes are active, utilize 601 bytes of memory, and 25 minutes have elapsed since its last change in status occurred.

To show a list of the alarm probes, enter:

```
-> show rmon probes alarm
```

A display showing all current alarm RMON probes should appear, as shown in the following example:

```
Entry   Slot/Port    Flavor      Status      Duration      System Resources

-------+-----------+-----------+----------+---------------+---------------------
4001    4/1          Alarm       Active      00:25:00      275 bytes
```

Displaying Statistics for a Particular RMON Probe

To view statistics for a particular current RMON probe, enter the **show rmon probes** command, specifying an entry number for a particular probe, such as:

```
-> show rmon probes 4005
```

A display showing statistics for the specified RMON probe will appear, as shown in the following sections.

Sample Display for Ethernet Statistics Probe

The display shown here identifies RMON Probe 4005’s Owner description and interface location (OmniSwitch Auto Probe on slot 4, port 5), Entry number (4005), probe Flavor (Ethernet statistics), Status (Active). Additionally, the display indicates the amount of time that has elapsed since the last change in status (48 hours, 54 minutes), and the amount of memory allocated to the probe, measured in bytes (275).

```
-> show rmon probes 4005

Probe’s Owner: OmniSwitch Auto Probe on Slot 4, Port 5
        Entry 4005
        Flavor = Ethernet, Status = Active
        Time = 48 hrs 54 mins,

        System Resources (bytes) = 275
```

Sample Display for History Probe

The display shown here identifies RMON Probe 10325's Owner description and interface location (Analyzer-p:128.251.18.166 on slot 4, port 5), the total number of History Control Buckets (samples) requested and granted (2), along with the time interval for each sample (30 seconds) and system-generated Sample Index ID number (5859). The probe Entry number identifier (10325), probe Flavor (History), and Status (Active), as well as the amount of time that has elapsed since the last change in status (48 hours, 53 minutes), and the amount of memory allocated to the probe, measured in bytes (601) are also displayed.

```
-> show rmon probes history 10325

Probe's Owner: Analyzer-p:128.251.18.166 on Slot 4, Port 5

          History Control Buckets Requested     = 2
          History Control Buckets Granted       = 2
          History Control Interval              = 30 seconds
          History Sample Index                  = 5859
          Entry 10325
                Flavor = History,
                Status = Active
                Time = 48 hrs 53 mins,
                System Resources (bytes) = 601
```

Sample Display for Alarm Probe

The display shown here identifies RMON Probe 11235's Owner description and interface location (Analyzer-t:128.251.18.166 on slot 4, port 8), as well as the probe's Alarm Rising Threshold and Alarm Falling Threshold, maximum allowable values beyond which an alarm will be generated and sent to the Event group (5 and 0, respectively).

Additionally, the corresponding Alarm Rising Event Index number (26020) and Alarm Falling Event Index number (0), which link the Rising Threshold Alarm and Falling Threshold Alarm to events in the Event table, are identified. The Alarm Interval, a time period during which data is sampled (10 seconds) and Alarm Sample Type (delta value—variable) are also shown, as is the Alarm Variable ID number (1.3.6.1.2.1.16.1.1.1.5.4008). The probe Entry number identifier (11235), probe Flavor (Alarm), and Status (Active), as well as the amount of time that has elapsed since the last change in status (48 hours, 48 minutes), and the amount of memory allocated to the probe, measured in bytes (1677) are also displayed.

```
-> show rmon probes alarm 11235

Probe's Owner: Analyzer-t:128.251.18.166 on Slot 4, Port 8
   Alarm Rising Threshold      = 5
   Alarm Falling Threshold     = 0
   Alarm Rising Event Index    = 26020
   Alarm Falling Event Index   = 0
   Alarm Interval              = 10 seconds
   Alarm Sample Type           = delta value
   Alarm Startup Alarm         = rising alarm
   Alarm Variable              = 1.3.6.1.2.1.16.1.1.1.5.4008
     Entry 11235
         Flavor = Alarm, Status = Active
         Time = 48 hrs 48 mins,
         System Resources (bytes) = 1677
```

Displaying a List of RMON Events

RMON Events are actions that occur based on Alarm conditions detected by an RMON probe. To view a list of logged RMON Events, enter the **show rmon events** command without specifying an entry number for a particular probe, such as:

```
-> show rmon events
```

A display showing all logged RMON Events should appear, as shown in the following example:

```
Entry      Time            Description

---------+----------------+-----------------------------------------------------
1          00:08:00        etherStatsPkts.4008: [Falling trap] “Falling Event”
2          00:26:00        etherStatsCollisions.2008: “Rising Event”
3          00:39:00        etherStatsCollisions.2008: “Rising Event”
```

The display shown above identifies the Entry number of the specified Event, along with the elapsed time since the last change in status (measured in hours/minutes/seconds) and a description of the Alarm condition detected by the probe for all RMON Logged Events. For example, Entry number 3 is linked to etherStatsCollisions.2008: [Rising trap] “Rising Event,” an Alarm condition detected by the RMON probe in which a trap was generated based on a Rising Threshold Alarm, with an elapsed time of 39 minutes since the last change in status.

Displaying a Specific RMON Event

To view information for a specific logged RMON Event, enter the **show rmon events** command, specifying an entry number (event number) for a particular probe, such as:

```
-> show rmon events 3
```

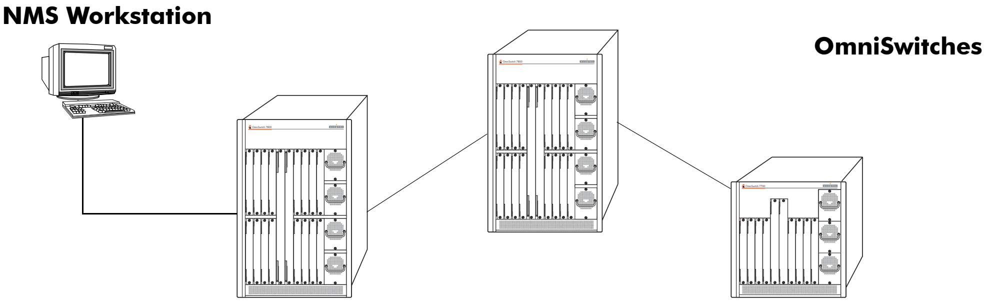
A display showing the specific logged RMON Event should appear, as shown in the following example:

```
Entry      Time            Description

---------+----------------+-----------------------------------------------------
3          00:39:00        etherStatsCollisions.2008: “Rising Event”
```

The display shown above identifies the Entry number of the specified Event, along with the elapsed time since the last change in status (measured in hours/minutes/seconds) and a description of the Alarm condition detected by the probe for the specific RMON Logged Event. For example, Entry number 3 is linked to etherStatsCollisions.2008: [Rising trap] “Rising Event,” an Alarm condition detected by the RMON probe in which a trap was generated based on a Rising Threshold Alarm, with an elapsed time of 39 minutes since the last change in status.

Monitoring Switch Health

To monitor resource availability, the NMS (Network Management System) needs to collect significant amounts of data from each switch. As the number of ports per switch (and the number of switches) increases, the volume of data can become overwhelming. The Health Monitoring feature can identify and monitor a switch's resource utilization levels and thresholds, improving efficiency in data collection.



Monitoring Resource Availability from Multiple Ports and Switches

Health Monitoring provides the following data to the NMS:

- Switch-level Input/Output, Memory and CPU Utilization Levels
- Module-level and Port-level Input/Output Utilization Levels

For each monitored resource, the following variables are defined:

- Most recent utilization level (percentage)
- Average utilization level over the last minute (percentage)
- Average utilization level over the last hour (percentage)
- Maximum utilization level over the last hour (percentage)
- Threshold level

Additionally, Health Monitoring provides the capacity to specify thresholds for the resource utilization levels it monitors, and generates traps based on the specified threshold criteria.

The following sections include a discussion of CLI commands that can be used to configure resource parameters and monitor or reset statistics for switch resources. These commands include:

- **health threshold**—Configures threshold limits for input traffic (RX), output/input traffic (TX/RX), memory usage**,** CPU usage, and chassis temperature. See page 29-25 for more information.
- **show health threshold**—Displays current health threshold settings. See page 29-26 for details.
- **health interval**—Configures sampling interval between health statistics checks. See page 29-27 for more information.
- **show health interval**—Displays current health sampling interval, measured in seconds. See page 29-27 for details.
- **show health** —Displays health statistics for the switch, as percentages of total resource capacity. See page 29-28 for more information.
- **health statistics reset**—Resets health statistics for the switch. See page 29-29 for details.

Configuring Resource and Temperature Thresholds

Health Monitoring software monitors threshold levels for the switch's consumable resources—*bandwidth, RAM memory, and CPU capacity*—as well as the ambient chassis temperature. When a threshold is exceeded, the Health Monitoring feature sends a trap to the Network Management Station (NMS). A trap is an alarm alerting the user to specific network events and, in the case of Health-related traps, indicates specifically which threshold has been crossed.

Note. When a resource falls back below the configured threshold, an addition trap is sent to the user. This indicates that the resource is no longer operating beyond its configured threshold limit.

The **health threshold** command is used to configure threshold limits for input traffic (RX), output/input traffic (TX/RX), memory usage, CPU usage and chassis temperature.

To configure thresholds for these resources, enter the **health threshold** command, followed by the input traffic, output/input traffic, memory usage, CPU usage, or chassis temperature value, where:

rx	Specifies an **input traffic** (RX) threshold, in percent. This value defines the maximum percentage of total bandwidth allowed for *incoming traffic only*. The total bandwidth is the Ethernet port capacity of *all NI modules* currently operating in the switch, in Mbps. For example, a chassis with 48 100Base-T Ethernet ports installed has a total bandwidth of 4800 Mbps. Since the default RX threshold is 80 percent, the threshold is exceeded if the input traffic on all ports reaches 3840 Mbps or higher.
txrx	Specifies a value for the **output/input traffic** (TX/RX) threshold. This value defines the maximum percentage of total bandwidth allowed for *all incoming and outgoing traffic*. As with the RX threshold described above, the total bandwidth is defined as the Ethernet port capacity for all NI modules currently operating in the switch, in Mbps. The default TX/RX threshold is 80 percent.
memory	Specifies a value for the **memory usage** threshold. Memory usage refers to the total amount of RAM memory currently used by switch applications. The default memory usage threshold is 80 percent.
cpu	Specifies a value for the **CPU usage** threshold. CPU usage refers to the total amount of CPU processor capacity currently used by switch applications. The default CPU usage threshold is 80 percent.
temperature	Specifies a value for the **chassis temperature** threshold (Celsius). The default temperature threshold is 50 degrees Celsius.

For example, to specify a CPU usage threshold of 85 percent, enter the following command:

```
-> health threshold cpu 85
```

For more information on the **health threshold** command, refer to Chapter 1, "Health Monitoring Commands," in the *OmniSwitch CLI Reference Guide*.

Note. When you specify a new value for a threshold limit, the value is automatically applied across all levels of the switch (switch, module and port). You cannot select differing values for each level.

Displaying Health Threshold Limits

The **show health threshold** command is used to view all current health thresholds on the switch, as well as individual thresholds for input traffic (RX), output/input traffic (TX/RX), memory usage, CPU usage and chassis temperature.

To view all health thresholds, enter the following command:

```
-> show health threshold
Rx Threshold            = 80
TxRx Threshold          = 80
Memory Threshold        = 80
CPU Threshold           = 80
Temperature Threshold   = 50
```

To display a specific health threshold, enter the **show health threshold** command, followed by the appropriate suffix syntax:

- **rx**
- **txrx**
- **memory**
- **cpu**
- **temperature**

For example, if you want to view only the health threshold for memory usage, enter the following command:

```
-> show health threshold memory
Memory Threshold      = 80
```

Note. For detailed definitions of each of the threshold types, refer to "Configuring Resource and Temperature Thresholds" on page 29-25, as well as Chapter 1, "Health Monitoring Commands," in the *OmniSwitch CLI Reference Guide*.

Configuring Sampling Intervals

The **sampling interval** is the period of time between polls of the switch's consumable resources to monitor performance vis-a-vis previously specified thresholds. The **health interval** command can be used to configure the sampling interval between health statistics checks.

To configure the sampling interval, enter the **health interval** command, followed by the number of seconds.

For example, to specify a **sampling interval** value of 6 seconds, enter the following command:

```
-> health interval 6
```

Valid values for the seconds parameter include 1, 2, 3, 4, 5, 6, 10, 12, 15, 20, or 30.

Note. If the sampling interval is decreased, switch performance may be affected.

Viewing Sampling Intervals

The **show health interval** command can be used to display the current health sampling interval (period of time between health statistics checks), measured in seconds.

To view the sampling interval, enter the **show health interval** command. The currently configured health sampling interval (measured in seconds) will be displayed, as shown below:

```
-> show health interval

Sampling Interval = 5
```

Viewing Health Statistics for the Switch

The **show health** command can be used to display health statistics for the switch.

To display health statistics, enter the **show health** command, followed by the slot/port location and optional **statistics** keyword.

For example, to view health statistics for the entire switch, enter the **show health** command without specifying any additional parameters. A screen similar to the following example will be displayed, as shown below:

```
-> show health
* - current value exceeds threshold

Device                          1 Min  1 Hr  1 Hr
Resources          Limit   Curr   Avg    Avg   Max
-----------------+-------+------+------+-----+----
Receive              80     00     00     00    00
Transmit/Receive     80     00     00     00    00
Memory               80     87*    87     86    87
Cpu                  80     08     05     04    08
Temperature Cmm      50     34     34     33    34
Temperature Cmm Cpu  50     28     28     27    28
```

In the screen sample shown above, the Device Resources field displays the device resources that are being measured (for example, Receive displays statistics for traffic received by the switch; Transmit/Receive displays statistics for traffic transmitted and received by the switch; Memory displays statistics for switch memory; and CPU displays statistics for the switch CPU). The Limit field displays currently configured device threshold levels as percentages of available bandwidth. The Curr field displays current bandwidth usage for the specified device resource. 1 Min. Avg. refers to the average device bandwidth used over a 1 minute period. 1 Hr. Avg. refers to the average device bandwidth used over a 1 hour period, and 1 Hr. Max. refers to the maximum device bandwidth used over a 1 hour period.

Note. If the Current value appears with an asterisk displayed next to it, the Current value exceeds the Threshold limit. For example, if the Current value for Memory displays as 85* and the Threshold Limit displays as 80, the asterisk indicates that the Current value has exceeded the Threshold Limit value.

Viewing Health Statistics for a Specific Interface

To view health statistics for slot 4/port 3, enter the **show health** command, followed by the appropriate slot and port numbers. A screen similar to the following example will be displayed, as shown below:

```
-> show health 4/3
* - current value exceeds threshold

Port 04/03                               1 Min      1 Hr       1 Hr
Resources            Limit    Curr       Avg        Avg        Max
------------------+----------+--------+--------+---------+--------------------
Receive              80       01         01         01         01
Transmit/Receive     80       01         01         01         01
```

In the screen sample shown above, the port 04/03 Resources field displays the port resources that are being measured (for example, Receive displays statistics for traffic received by the switch, while Transmit/Receive displays statistics for traffic transmitted and received by the switch. The Limit field displays currently configured resource threshold levels as percentages of available bandwidth. The Curr field displays current bandwidth usage for the specified resource. 1 Min. Avg. refers to the average resource bandwidth used over a 1 minute period. 1 Hr. Avg. refers to the average resource bandwidth used over a 1 hour period, and 1 Hr. Max. refers to the maximum resource bandwidth used over a 1 hour period.

Resetting Health Statistics for the Switch

The **health statistics reset** command can be used to clear health statistics for the entire switch. This command cannot be used to clear statistics only for a specific module or port.

To reset health statistics for the switch, enter the **health statistics reset** command, as shown below:

```
-> health statistics reset
```

30 Using Switch Logging

The switch logging utility provides a high-level event logging mechanism that is useful in maintaining and servicing the switch. Switch logging uses a formatted string mechanism to either record or discard event data from switch applications. The log records are copied to the output devices configured for the switch. Log records can be sent to a text file written into the flash file system. The log records can also be scrolled to the switch's console or to a remote IP address.

Switch logging information can be customized and configured through Command Line Interface (CLI) commands, WebView, and SNMP. Log information can be helpful in resolving configuration or authentication issues as well as general switch errors.

This chapter describes the switch logging feature; how to configure it and display switch logging information through the Command Line Interface (CLI). CLI commands are used in the configuration examples. For more details about the syntax of commands, see the *OmniSwitch CLI Reference Guide*.

In This Chapter

The following procedures are described:

- "Enabling Switch Logging" on page 30-6
- "Setting the Switch Logging Severity Level" on page 30-6
- "Specifying Switch Logging Output Device" on page 30-9
- "Displaying Switch Logging Status" on page 30-10
- "Displaying Switch Logging Records" on page 30-12

Notes. The switch logging commands are not intended for use with low-level hardware and software debugging. It is strongly recommended that you contact an Alcatel Customer Service representative for assistance with debugging functions.

Switch Logging Specifications

Functionality Supported	High-level event logging mechanism that forwards requests from applications to enabled logging devices.
Functionality Not Supported	Not intended for debugging individual hardware applications
Logging Devices Supported	Flash Memory/Console/IP Address
Application ID Levels/Numeric Equivalents Supported	IDLE (255), DIAG (0), IPC-DIAG (1), QDRIVER (2), QDISPATCHER (3), IPC-LINK (4), NI-SUPERVISION (5), INTERFACE (6), 802.1Q (7), VLAN (8), GM (9), BRIDGE (10), STP (11), LINKAGG (12), QOS (13), RSVP (14), IP (15), IPMS (17), AMAP (18), GMAP (19), AAA (20), IPC-MON (21), IP-HELPER (22), PMM (23), MODULE (24), SLB (25), EIPC (26), CHASSIS (64), PORT-MGR (65), CONFIG (66), CLI (67), SNMP (68), WEB (69), MIPGW (70), SESSION (71), TRAP (72), POLICY (73), DRC (74), SYSTEM (75), HEALTH (76), VRRP (77), NAN-DRIVER (78), RMON (79), TELENET (80), PSM (81), FTP (82), SNMI (83), DISTRIB (84), EPIL0GUE (85), LDAP (86), NOSNMP (87), SSL (88), DBGGW (89), LANPOWER (108)
Severity Levels/Types Supported	2 (Alarm - highest severity), 3 (Error), 4 (Alert), 5 (Warning) 6 (Info - default), 7 (Debug 1), 8 (Debug 2), 9 (Debug 3 - lowest severity)

Switch Logging Defaults

The following table shows switch logging default values.

Global Switch Logging Defaults

Parameter Description	CLI Command	Default Value/Comments
Enabling/Disabling switch logging	**swlog**	Enabled
Switch logging severity level	**swlog appid level**	No application ID or severity level defaults. The user must specify these values
Enabling/Disabling switch logging Output	**swlog output**	Flash Memory and Console
Switch logging file size	**swlog output flash file-size**	128000 bytes

Quick Steps for Configuring Switch Logging

1 Enable switch logging by using the following command:

```
-> swlog
```

2 Specify the ID of the application to be logged along with the logging severity level.

```
-> swlog appid slb level warning
```

Here, the application ID specifies server load balancing (slb) and severity is set to the “warning” level.

3 Specify the output device to which the switch logging information will be sent.

```
-> swlog output console
```

In this example, the switch logging information will be sent to the console port.

Note. *Optional.* To verify the switch logging configuration, enter the **show swlog** command. The display is similar to the one shown below:

```
Switch Logging is:
             - INITIALIZED
             - RUNNING

Log Device(s)
----------------
flash
console

Only Applications not at the level ‘info’ (6) are shown
Application ID  Level
----------------------------
SLB(25)         warning (5)
```

For more information about this command, see the “Switch Logging Commands” chapter in the *OmniSwitch CLI Reference Guide*.

Switch Logging Overview

The switch logging utility is an event logging mechanism used to monitor switch operations. Switch logging uses a formatted string mechanism to process log requests from applications. When a log request is received, switch logging compares the severity level included with the request to the severity level stored for the application ID. If there is a match, a log message is generated using the format specified by the log request and placed on the switch log queue. Switch logging then returns control back to the calling application.

You can specify the path to where the log file will be printed in the switch's flash file system. You can also send the log file to other output devices, such as the console or remote IP address. In this case, the log records generated are copied to all configured output devices.

Switch logging information can be displayed and configured through CLI commands, WebView and SNMP. The information generated by switch logging can be helpful in resolving configuration or authentication issues, as well as general errors.

Notes. Although switch logging provides complementary functionality to switch debugging facilities, the switch logging commands are not intended for use with low-level hardware and software debugging functions.

The **configuration snapshot** command can be used to capture and save all switch logging configuration settings in a text file that can be viewed, edited, and used as a configuration file. See the “Working with Configuration Files” chapter of the *OmniSwitch 7700/7800/8800 Switch Management Guide* for details.

Switch Logging Commands Overview

This section describes the switch logging CLI commands, for enabling or disabling switch logging, displaying the current status of the switch logging feature, and displaying stored log information.

Enabling Switch Logging

The **swlog** command initializes and enables switch logging, while **no swlog** disables it.

To enable switch logging, enter the **swlog** command:

```
-> swlog
```

To disable switch logging, enter the **no swlog** command:

```
-> no swlog
```

No confirmation message will appear on the screen for either command.

Setting the Switch Logging Severity Level

The switch logging feature can log all switch error-type events for a particular switch application. You can also assign severity levels to the switch applications that will cause some of the events to be filtered out of your display. The **swlog appid level** command is used to assign the severity levels to the applications.

The syntax for the **swlog appid level** command requires that you identify a switch application and assign it a severity level. The severity level setting controls the kinds of error-type events that will be recorded by the switch logging function. If an application experiences an event equal to or greater than the severity level assigned to the application, the event will be recorded and forwarded to the configured output devices. You can specify the application either by the application ID CLI keyword or by its numeric equivalent.

The application ID information is shown in the following table. The severity level information is shown in the table beginning on page 30-8.

CLI Keyword	**Numeric Equivalent**	**Application ID**
IDLE	**255**	APPID_IDLE
DIAG	**0**	APPID_DIAGNOSTICS
IPC-DIAG	**1**	APPID_IPC_DIAGNOSTICS
QDRIVER	**2**	APPID_QDRIVER
QDISPATCHER	**3**	APPID_QDISPATCHER
IPC-LINK	**4**	APPID_IPC_LINK
NI-SUPERVISION	**5**	APPID_NI_SUP_AND_PROBER
INTERFACE	**6**	APPID_ESM_DRIVER
802.1Q	**7**	APPID_802.1Q
VLAN	**8**	APPID_VLAN_MGR
GM	**9**	APPID_GROUPMOBILITY (RESERVED)

CLI Keyword	Numeric Equivalent	Application ID
BRIDGE	**10**	APPID_SRCLEANING
STP	**11**	APPID_SPANNINGTREE
LINKAGG	**12**	APPID_LINKAGGREGATION
QOS	**13**	APPID_QOS
RSVP	**14**	APPID_RSVP
IP	**15**	APPID_IP
IPX	**16**	APPID_IPX
IPMS	**17**	APPID_IPMS
AMAP	**18**	APPID_XMAP
GMAP	**19**	APPID_GMAP
AAA	**20**	APPID_AAA
IPC-MON	**21**	APPID_IPC_MON
IP-HELPER	**22**	APPID_BOOTP_RELAY
PMM	**23**	APPID_MIRRORING_MONITORING
MODULE	**24**	APPID_L3HRE
SLB	**25**	APPID_SLB
EIPC	**26**	APPID_EIPC
CHASSIS	**64**	APPID_CHASSISUPER
PORT-MGR	**65**	APPID_PORT_MANAGER
CONFIG	**66**	APPID_CONFIGMANAGER
CLI	**67**	APPID_CLI
SNMP	**68**	APPID_SNMP_AGENT
WEB	**69**	APPID_WEBMGT
MIPGW	**70**	APPID_MIPGW
SESSION	**71**	APPID_SESSION_MANAGER
TRAP	**72**	APPID_TRAP_MANAGER
POLICY	**73**	APPID_POLICY_MANAGER
DRC	**74**	APPID_DRC
SYSTEM	**75**	APPID_SYSTEM_SERVICES
HEALTH	**76**	APPID_HEALTHMON
VRRP	**77**	APPID_VRRP
NAN-DRIVER	**78**	APPID_NAN_DRIVER
RMON	**79**	APPID_RMON
TELNET	**80**	APPID_TELNET
PSM	**81**	APPID_PSM

CLI Keyword	Numeric Equivalent	Application ID
FTP	**82**	APPID_FTP
SMNI	**83**	APPID_SMNI
DISTRIB	**84**	APPID_DISTRIB
EPILOGUE	**85**	APPID_EPILOGUE
LDAP	**86**	APPID_LDAP
NOSNMP	**87**	APPID_NOSNMP
SSL	**88**	APPID_SSL
DBGGW	**89**	APPID_DBGGW
LANPOWER	**108**	APPID_LANPOWER

The **level** keyword assigns the error-type severity level to the specified application IDs. Values range from 2 (highest severity) to 9 (lowest severity). The values are defined in the following table:

Severity Level	Type	Description
2 *(highest severity)*	**Alarm**	A serious, non-recoverable error has occurred and the system should be rebooted.
3	**Error**	System functionality is reduced.
4	**Alert**	A violation has occurred.
5	**Warning**	An unexpected, non-critical event has occurred.
6 *(default)*	**Info**	Any other non-debug message.
7	**Debug 1**	A normal event debug message.
8	**Debug 2**	A debug-specific message.
9 *(lowest severity)*	**Debug 3**	A maximum verbosity debug message.

Specifying the Severity Level

To specify the switch logging severity level, use the **swlog appid level** command. The application ID can be expressed by using either the ID number or the application ID CLI keyword as listed in the table beginning on page 30-6. The severity level can be expressed by using either the severity level number or the severity level type as shown in the table above. The following syntax assigns the “warning” severity level (or 5) for the “system” application, (ID number 75) by using the severity level and application names.

```
-> swlog appid system level warning
```

The following command makes the same assignment by using the severity level and application numbers.

```
-> swlog appid 75 level 3
```

No confirmation message appears on the screen for either command.

Removing the Severity Level

To remove the switch logging severity level, enter the **no swlog appid level** command, including the application ID and severity level values. The following is a typical example:

```
-> no swlog appid 75 level 5
```

Or, alternatively, as:

```
-> no swlog appid system level warning
```

No confirmation message will appear on the screen.

Specifying Switch Logging Output Device

The **swlog output** command allows you to send the switch logging information to your console, to the switch's flash memory or to a specified IP address(es).

Enabling/Disabling Switch Logging Output to the Console

To enable the switch logging output to the console, enter the following command:

```
-> swlog output console
```

To disable the switch logging output to the console, enter the following command:

```
-> no swlog output console
```

No confirmation message will appear on the screen for either command.

Enabling/Disabling Switch Logging Output to Flash Memory

To enable the switch logging output to flash memory, enter the following:

```
-> swlog output flash
```

To disable the switch logging output to flash memory, enter the following command:

```
-> no swlog output flash
```

No confirmation message will appear on the screen for either command.

Specifying an IP Address for Switch Logging Output

To specify a particular IP address destination (e.g., a server) for switch logging output, enter the **swlog output socket ipaddr** command, specifying the target IP address to which output will be sent. For example, if the target IP address is 168.23.9.100, you would enter:

```
-> swlog output socket ipaddr 168.23.9.100
```

No confirmation message will appear on the screen.

Note. You can also send syslog files to multiple hosts (maximum of four).

Disabling an IP Address from Receiving Switch Logging Output

To disable a particular IP address from receiving switch logging output, enter the following command:

```
-> no swlog output socket
```

No confirmation message will appear on the screen.

Note. It is not necessary to specify the IP address in the **no swlog output socket** command.

Displaying Switch Logging Status

You can display the current status of switch logging onto your console screen by using the **show swlog** command. The following information is displayed:

- The enable/disable status of switch logging.
- A list of current output devices configured for switch logging.
- The switch logging severity level for each application that is not set to the “info” (6) setting.

The following is a sample display:

```
-> show swlog

Switch Logging is:
              - INITIALIZED
              - RUNNING

Log Device(s)
-------------
flash
console
socket ipaddr 11.1.1.1
socket ipaddr 12.1.1.1
socket ipaddr 13.1.1.1
socket ipaddr 14.1.1.1

All Applications have their trace level set to the level 'info' (6)
```

For this example, switch logging is enabled. Switch logging information is being sent to the switch’s flash memory and to the console. Additionally, the severity level for the chassis application ID has been set to the “debug3” (or “9”) severity level.

Configuring the Switch Logging File Size

By default, the size of the switch logging file is 128000 bytes. To configure the size of the switch logging file use the **swlog output flash file-size** command. To use this command enter **swlog output flash file size** followed by the number of bytes, which must be at least 32000. (The maximum size the file can be is dependent on the amount of free memory available in flash memory.)

Note. Use the **ls** command, which is described in the *OmniSwitch 7700/7800/8800 Switch Management Guide*, to determine the amount of available flash memory.

For example, to set the switch logging file to 500000 bytes enter:

```
-> swlog output flash file-size 500000
```

Clearing the Switch Logging Files

You can clear the data stored in the switch logging files by executing the following command:

```
-> swlog clear
```

This command will cause the switch to clear all switch logging information and begin recording again. As a result, the switch will display a shorter file when you execute the **show log swlog** command. You may want to use **swlog clear** when the switch logging display is too long due to some of the data being old or out dated.

No confirmation message will appear on the screen.

Displaying Switch Logging Records

The **show log swlog** command can produce a display showing *all* switch logging information or you can display information according to session, timestamp, application Id or severity level. For details refer to the *OmniSwitch CLI Reference Guide*. The following sample screen output shows a display of all switch logging information.

Note. Switch logging frequently records a very large volume of data. It can take several minutes for all switch logging information to scroll down the console screen.

```
-> show log swlog
Displaying file contents for 'swlog2.log'
FILEID: fileName[swlog2.log], endPtr[32]
         configSize[64000], currentSize[64000], mode[2]
Displaying file contents for 'swlog1.log'
FILEID: fileName[swlog1.log], endPtr[395]
         configSize[64000], currentSize[64000], mode[1]

Time Stamp               Application    Level   Log Message

------------------------+--------------+-------+--------------------------------

MON NOV 11 12:42:11 2002         SYSTEM    info Switch Logging files cleared by
command
MON NOV 11 13:07:26 2002            WEB    info The HTTP session login successfu
l!
MON NOV 11 13:18:24 2002            WEB    info The HTTP session login successfu
l!
MON NOV 11 13:24:03 2002         TELNET    info New telnet connection, Address,
128.251.30.88
MON NOV 11 13:24:03 2002         TELNET    info Session 4, Created
MON NOV 11 13:59:04 2002            WEB    info The HTTP session user logout suc
cessful!
```

The fields in the above example are defined as follows:

- The **FILE ID** field specifies the File name (e.g., swlog1.log), endPtr Global Sequence ID reference number (e.g., 9968), Configuration Size (e.g., 10000), Current Size (e.g., 10000), and Mode (e.g., 2).
- The **Timestamp** field indicates when the swlog entry occurred (e.g., THU, NOV 12, 02:06:52 2001).
- The **Application** field specifies the application ID for which the stored swlog information is displayed (e.g., SYSTEM).
- The **Level** field specifies the severity level for which the stored information is displayed (e.g., Warning).
- The **Log Message** field specifies the condition recorded by the switch logging feature. The information in this field usually wraps around to the next line of the screen display as shown in this example.

31 Monitoring Memory

Debug memory monitor commands can monitor memory allocation and free memory (such as detection of invalid free addresses and maintenance of size statistics). These commands are useful for monitoring logging of events, leak detection, classification of memory allocations, detection of invalid free addresses, and maintenance of size statistics.

Notes. System Debug (kTrace and sysTrace) commands are intended for use by qualified Alcatel Customer Support personnel to assist customers in diagnosing or debugging system performance. For information about these commands, see the chapter titled, "Memory Monitoring Commands" in the *OmniSwitch CLI Reference Guide.* It is strongly recommended that you contact an Alcatel Customer Service representative for assistance.

The Switch Logging feature is a high-level event logging mechanism that can also be useful in maintaining and servicing the switch. For information about this feature, see Chapter 30, "Using Switch Logging".

The **configuration snapshot** command can be used to capture and save all kTrace, sysTrace, Memory Monitor, and Switch Logging configuration settings in a "snapshot" text file that can be viewed, edited, and used as a configuration file. See the chapter titled, "Working with Configuration Files" in your *OmniSwitch 7700/7800/8800 Switch Management Guide* for details.

In This Chapter

This chapter describes the Memory Monitoring Commands and how to configure and display them in the Command Line Interface (CLI). CLI commands are used in the examples. For more information about the syntax of commands, see the *OmniSwitch CLI Reference Guide*.

The procedures described in this chapter include:

- "Enabling/Disabling Memory Monitoring Functions" on page 31-4.
- "Displaying the Memory Monitor Log" on page 31-5.
- "Displaying the Memory Monitor Global Statistics" on page 31-6.
- "Displaying the Memory Monitor Task Statistics" on page 31-7.
- "Displaying the Memory Monitor Size Statistics" on page 31-9.

Memory Monitoring Specifications

The following table shows Memory Monitoring specifications:

Functionality Supported	Fence Post/ Bad Address Detection/ Leak Monitoring/ Memory Classification/ Global Statistical Gathering/ Task Statistical Gathering/ Size Statistical Gathering.
Functionality Not Supported	Ownership Violations.
Show Command Output Devices Supported	Standard Out (console)/ Switch Logging/ sysTrace Buffer.

Memory Monitoring Defaults

The following table shows Memory Monitoring default values:

Parameter Description	**CLI Command**	**Default Value/Comments**
Memory Monitoring	**debug memory monitor**	Disabled

Quick Steps for Configuring Memory Monitoring

1 Use the following commands to enable Memory Monitoring. (Memory Monitoring is factory disabled by default.) For example:

```
-> debug memory monitor enable
```

2 To view Memory Monitoring log information, enter the **debug memory monitor show log** command. The display is similar to the one shown below:

```
-> debug memory monitor show log

   Task             Memory Memory Addr of OS func   Calling    Previous
   Name    Comments  Addr   Size    OS call Called    Function    Caller
---------+--------+-------+------+---------+---------+----------+--------------
tssApp_2* TCB Stac 00ca1550 20680  0013a180 objAllocEx taskSpawn ssAppChild
tssApp_2* Vx B Sem 02317ca8   28   001374d0 objAlloc   pipe      ssAppChild
tssApp_2* Vx B Sem 02317f78   28   001374d0 objAlloc   pipe      ssAppChild
tssApp_2*          0107be78  5121  0012cfc8 malloc     pipe       ssAppChild
tssApp_2*          023182b0   16   0012cfa8 malloc     pipe       ssAppChild
tssApp_2*          024fdc90    9   00105fb0 malloc     pipe       ssAppChild
tssApp_2*          016d6548  288   000af228 malloc     ssAppChild mip_msg_qu
CliShell0 Vx C Sem 035fe590   28   0011f038 semCCreate zcSelect   mip_msg_do
```

The information displayed above includes the task that owns the memory block, the type of memory block that has been allocated, the address and size of the memory block, the address of the operating system call that allocated the memory block, the function that contained the call that allocated the memory block, and the Calling Function and Previous Caller function (that called the above-mentioned functions).

Note. *Optional.* Display the Memory Monitoring Size Statistics. The display is similar to the one shown below.

```
-> debug memory monitor show log size

Lower Upper   Currently   Cummulatively
Limit Limit   Allocated     Allocated
-----+-----+-------------+-------------+
    0    16        14439         31689
   16    32         6299       7704923
   32    64         4833        373109
   64   128        44248        145775
  128   256        12367        122315
  256   512        52096        228673
  512  1024        26778        365552
 1024  2048        24572        358630
 2048  4096        49648        274071
 4096  8192        50793       1534291
 8192 16384       478292        673610
16384 32768       431784       1075783
32768 65536       850216       1588017
65536            5130020       25675316
```

For more information about this command, see "Displaying the Memory Monitor Log" on page 31-5 or the "Switch Logging Commands" chapter in the *OmniSwitch CLI Reference Guide.*

Debug Memory Commands Overview

The Debug Memory Commands provide monitoring of memory allocation and free memory. By providing a method to enable/disable memory monitoring and display memory usage reports, these commands can be used to monitor logging of events, leak detection, classification of memory allocations, detection of invalid free addresses, and maintenance of size statistics. Additionally, the following automatic outputs will occur under specific conditions:

- If there is an attempt to free an invalid address, the monitor will create a switch log message, cause a Post Mortem Dump (PMD) of the memory monitor variables and log, and suspend the task. For information about using the **show log pmd** command to view the contents of a stored PMD file, see the chapter titled, "Memory Monitoring Commands" in the *OmniSwitch CLI Reference Guide.*

- If a memory leak of unclassified memory is detected, the service will generate a sysTrace (System Trace) message. The system trace facility provides a consistent, high-level mechanism for capturing event records in a history buffer. Captured sysTrace information can be referenced for system debugging. For information about using the sysTrace utility to enable, disable or view sysTrace log information, see the chapter titled, "Memory Monitoring Commands" in the *OmniSwitch CLI Reference Guide.*

Configuring Debug Memory Commands

This section describes the Debug Memory commands, which include separate commands for enabling or disabling memory monitoring, as well as displaying memory log information, global statistics, task statistics, and size statistics.

Enabling/Disabling Memory Monitoring Functions

The **debug memory monitor** command enables or disables the memory monitoring functions.

To enable memory monitoring, enter:

```
-> debug memory monitor enable
```

No confirmation message will appear onscreen.

To disable Memory Monitoring, enter:

```
-> debug memory monitor disable
```

No confirmation message will appear onscreen.

Displaying the Memory Monitor Log

The **debug memory monitor show log** command displays memory monitoring log information. By entering this command, a display similar to the following will appear onscreen:

```
-> debug memory monitor show log

   Task            Memory Memory Addr of  OS func   Calling    Previous
   Name   Comments  Addr   Size  OS call   Called    Function    Caller
---------+--------+-------+-----+---------+----------+----------+------------------
tssApp_2* TCB Stac 00ca1550 20680 0013a180 objAllocEx taskSpawn  ssAppChild
tssApp_2* Vx B Sem 02317ca8    28 001374d0 objAlloc   pipe       ssAppChild
tssApp_2* Vx B Sem 02317f78    28 001374d0 objAlloc   pipe       ssAppChild
tssApp_2*          0107be78  5121 0012cfc8 malloc     pipe       ssAppChild
tssApp_2*          023182b0    16 0012cfa8 malloc     pipe       ssAppChild
tssApp_2*          024fdc90     9 00105fb0 malloc     pipe       ssAppChild
tssApp_2*          016d6548   288 000af228 malloc     ssAppChild mip_msg_qu
CliShell0 Vx C Sem 035fe590    28 0011f038 semCCreate zcSelect   mip_msg_do
SsApp     Vx C Sem 035fe4b8    28 0011f038 semCCreate zcSelect   tssAppMain
CliShell0          02318250     2 02b33a3c malloc     SSLexLexem SSYaccStac
CliShell0          02317538    56 02b33a3c malloc     SSLexLexem SSYaccStac
CliShell0          016d6670   272 02b33a3c malloc     SSYaccStac SSYaccPars
CliShell0          02318260     1 02b33a3c malloc     SSLexLexem SSYaccStac
CliShell0          02317718    56 02b33a3c malloc     SSLexLexem SSYaccStac
CliShell0          016d68b0   272 02b33a3c malloc     SSYaccStac PropagateP
CliShell0          023182c8     4 02b33a3c malloc     SSLexLexem SSYaccStac
CliShell0          027b0060    56 02b33a3c malloc     SSLexLexem SSYaccStac
CliShell0          01896b28   272 02b33a3c malloc     SSYaccStac SSYaccPars
CliShell0          023182d8     4 02b33a3c malloc     SSLexLexem SSYaccStac
CliShell0          035fe4e0    56 02b33a3c malloc     SSLexLexem SSYaccStac
CliShell0          01e3d928   272 02b33a3c malloc     SSYaccStac SSYaccPars
CliShell0          024fdca8     4 02b33a3c malloc     SSLexLexem SSYaccStac
CliShell0          035fe3e0    56 02b33a3c malloc     SSLexLexem SSYaccStac
CliShell0          022b3ab0   272 02b33a3c malloc     SSYaccStac SSYaccPars
CliShell0          024fdcb8     3 02b33a3c malloc     SSLexLexem SSYaccStac
CliShell0          01e37e40    56 02b33a3c malloc     SSLexLexem SSYaccStac
CliShell0          022b3bc8   272 02b33a3c malloc     SSYaccStac SSYaccPars
CliShell0          02314da8   272 02b33a3c malloc     SSYaccStac SSYaccInit
CliShell0          023183d8   512 02b33a3c malloc     CliParse   clishell_m
CliShell0          027b0100   576 02b33a3c malloc     CliParse   clishell_m
CliShell0          0107a128  2404 02b33a3c malloc     CliParse   clishell_m
CliShell0          0107aa98  1280 02b33a3c malloc     CliParse   clishell_m
Stp       Vx C Sem 024fdcc8    28 0011f038 semCCreate zcSelect   stpSock_st
LnkAgg    Vx C Sem 023182e8    28 0011f038 semCCreate zcSelect   lagg_Sock_
AmapMgr   Vx C Sem 02318270    28 0011f038 semCCreate zcSelect   xmap_main_
GrpMob    Vx C Sem 035fe5b8    28 0011f038 semCCreate zcSelect   gmcWaitFor
GmapMgr   Vx C Sem 02317fa0    28 0011f038 semCCreate zcRecvfrom gmap_main_
VlanMgr   Vx C Sem 02317cd0    28 0011f038 semCCreate zcSelect   vmcWaitFor
NanDrvr   Vx C Sem 02318158    28 0011f038 semCCreate zcRecvfrom nanDriver
```

In the screen sample shown above, the **Task Name** field displays the task that owns the memory block. The **Comments** field explains what type of memory block has been allocated. The **Memory Addr** and **Memory Size** fields display the address and size of the memory block. The **Addr of OS Call** field displays the address of the operating system call that allocated the memory block. The **OS func Called** field displays the function that contained the call that allocated the memory block. The **Calling Function** field displays the function that called the above-mentioned function. The **Previous Caller** field displays the function that called the above-mentioned function.

Displaying the Memory Monitor Global Statistics

The **debug memory monitor show log global** command can display memory monitoring global statistics. By specifying the global variable to view global statistics, a display similar to the following should appear:

```
-> debug memory monitor show log global

Current    = 33741
Cumulative = 687952
```

In the screen sample shown above, the **Current** and **Cumulative** fields display statistics indicating the amount of dynamic memory allocated globally (currently and cumulatively) since the memory log was last enabled. For example, statistics displayed above indicate that 33741 bytes of memory are currently allocated and 687952 bytes were cumulatively allocated since the last enable.

Displaying the Memory Monitor Task Statistics

The **debug memory monitor show log task** command can display memory monitoring task statistics. By specifying the task variable to view task statistics, a display similar to the following should appear:

```
-> debug memory monitor show log task

Task Name       Current        Cumulative
-------------+-------------+-----------------
tssApp0_4       26369          52594
cliConsole      16169          20186
tIpxGapper      242            242
tIpxTimer       214            214
tDrcIprm        1801287        1801315
DrcTm           479453         675448
WebView         53690          340083
Rmon            285084         334616
SlbCtrl         578            578
PolMgr          808            15704
Qos             47096          938852
UdpRly          8320           8348
Vrrp            622            1198
Ipx             29634          29634
ipmpm           231152         231152
ipmfm           480422         480450
Ipmem           423686         423686
GmapMgr         9128           263872
AmapMgr         284            891188
LnkAgg          86988          1867592
8021q           128            184
Ipx             29634          29634
stpTick         1024           1024
Stp             70782          1555454
GrpMob          128            669300
SrcLrn          12516          12572
EsmDrv          356            74752
PsMgr           168            308
L3Hre           528            528
Health          249            127649
AAA             221312         222236
Ipedr           31500          105868
NanDrvr         56             74396
Ftpd            56             56
Telnetd         9552           9552
tCS_CVM         28             28
tssApp65535_3   228            228
SsApp           49088          198284
SesMgr          69200          202029
SNMPagt         26347          210129
```

--Output continues on the following page--

```
Task Name       Current        Cumulative
-------------+-------------+-----------------
TrapMgr         4548           63976
Elpc            2336           2392
VlanMgr         208            149672
PortMgr         804            75424
Gateway         84             140
CfgMgr          228            897491
tCS_HSM         1240           2500
tCS_CMS         188            328
tCS_PRB         312            340
tCS_CCM         612            12555
tCSCSMtask      586128         15256874
tSwLogTask                     13519+

->
```

In the screen sample shown above, the **Task Name** field identifies the Task ID. The **Current** and **Cumulative** fields display statistics indicating the amount of dynamic memory allocated to the specified task (currently and cumulatively) since the memory log was enabled. For example, statistics displayed in the second entry in the table indicate that 16169 bytes of memory are currently allocated and 20168 bytes were cumulatively allocated for the **cliConsole** task.

Displaying the Memory Monitor Size Statistics

The **debug memory monitor show log size** command can display memory monitoring size statistics. By entering the size variable to view size statistics, a display similar to the following should appear:

```
-> debug memory monitor show log size
   Lower       Upper     Currently    Cumulatively
   Limit       Limit     Allocated     Allocated
------------+----------+------------+------------------
    0           16         14439       31689
    16          32         6299        7704923
    32          64         4833        373109
    64          128        44248       145775
    128         256        12367       122315
    256         512        52096       228673
    512         1024       26778       365552
    1024        2048       24572       358630
    2048        4096       49648       274071
    4096        8192       50793       1534291
    8192        16384      478292      673610
    16384       32768      431784      1075783
    32768       65536      850216      1588017
    65536                  5130020     25675316

->
```

In the screen sample shown above, the **Lower Limit** and **Upper Limit** fields display statistics indicating the lower limit and upper limit of the memory (range) being sampled. The **Currently Allocated** and **Cumulatively Allocated** fields display statistics indicating the amount of memory currently allocated and cumulatively allocated to the specified size ranges (in bytes). For example, statistics displayed in the last entry in the table indicate that 5130020 bytes are currently allocated and 25675316 bytes were cumulatively allocated for the memory range greater than or equal to 65536 bytes.

A Software License and Copyright Statements

This appendix contains Alcatel and third-party software vendor license and copyright statements.

Alcatel License Agreement

ALCATEL INTERNETWORKING, INC. (“AII”) SOFTWARE LICENSE AGREEMENT

IMPORTANT. Please read the terms and conditions of this license agreement carefully before opening this package.

By opening this package, you accept and agree to the terms of this license agreement. If you are not willing to be bound by the terms of this license agreement, do not open this package. Please promptly return the product and any materials in unopened form to the place where you obtained it for a full refund.

1. **License Grant.** This is a license, not a sales agreement, between you (the “Licensee”) and AII. AII hereby grants to Licensee, and Licensee accepts, a non-exclusive license to use program media and computer software contained therein (the “Licensed Files”) and the accompanying user documentation (collectively the “Licensed Materials”), only as authorized in this License Agreement. Licensee, subject to the terms of this License Agreement, may use one copy of the Licensed Files on the Licensee’s system. Licensee agrees not to assign, sublicense, transfer, pledge, lease, rent, or share their rights under this License Agreement. Licensee may retain the program media for backup purposes with retention of the copyright and other proprietary notices. Except as authorized under this paragraph, no copies of the Licensed Materials or any portions thereof may be made by Licensee and Licensee shall not modify, decompile, disassemble, reverse engineer, or otherwise attempt to derive the Source Code. Licensee is also advised that AII products contain embedded software known as firmware which resides in silicon. Licensee may not copy the firmware or transfer the firmware to another medium.

2. **AII’s Rights.** Licensee acknowledges and agrees that the Licensed Materials are the sole property of AII and its licensors (herein “its licensors”), protected by U.S. copyright law, trademark law, and are licensed on a right to use basis. Licensee further acknowledges and agrees that all rights, title, and interest in and to the Licensed Materials are and shall remain with AII and its licensors and that no such right, license, or interest shall be asserted with respect to such copyrights and trademarks. This License Agreement does not convey to Licensee an interest in or to the Licensed Materials, but only a limited right to use revocable in accordance with the terms of this License Agreement.

3. **Confidentiality.** AII considers the Licensed Files to contain valuable trade secrets of AII, the unauthorized disclosure of which could cause irreparable harm to AII. Except as expressly set forth herein, Licensee agrees to use reasonable efforts not to disclose the Licensed Files to any third party and not to use the Licensed Files other than for the purpose authorized by this License Agreement. This confidentiality obligation shall continue after any termination of this License Agreement.

4. **Indemnity.** Licensee agrees to indemnify, defend and hold AII harmless from any claim, lawsuit, legal proceeding, settlement or judgment (including without limitation AII's reasonable United States and local attorneys' and expert witnesses' fees and costs) arising out of or in connection with the unauthorized copying, marketing, performance or distribution of the Licensed Files.

5. **Limited Warranty.** AII warrants, for Licensee's benefit alone, that the program media shall, for a period of ninety (90) days from the date of commencement of this License Agreement (referred to as the Warranty Period), be free from defects in material and workmanship. AII further warrants, for Licensee benefit alone, that during the Warranty Period the Licensed Files shall operate substantially in accordance with the functional specifications in the User Guide. If during the Warranty Period, a defect in the Licensed Files appears, Licensee may return the Licensed Files to AII for either replacement or, if so elected by AII, refund of amounts paid by Licensee under this License Agreement. EXCEPT FOR THE WARRANTIES SET FORTH ABOVE, THE LICENSED MATERIALS ARE LICENSED "AS IS" AND AII AND ITS LICENSORS DISCLAIM ANY AND ALL OTHER WARRANTIES, WHETHER EXPRESS OR IMPLIED, INCLUDING (WITHOUT LIMITATION) ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. SOME STATES DO NOT ALLOW THE EXCLUSION OF IMPLIED WARRANTIES SO THE ABOVE EXCLUSIONS MAY NOT APPLY TO LICENSEE. THIS WARRANTY GIVES THE LICENSEE SPECIFIC LEGAL RIGHTS. LICENSEE MAY ALSO HAVE OTHER RIGHTS WHICH VARY FROM STATE TO STATE.

6. **Limitation of Liability.** AII's cumulative liability to Licensee or any other party for any loss or damages resulting from any claims, demands, or actions arising out of or relating to this License Agreement shall not exceed the license fee paid to AII for the Licensed Materials. IN NO EVENT SHALL AII BE LIABLE FOR ANY INDIRECT, INCIDENTAL, CONSEQUENTIAL, SPECIAL, OR EXEMPLARY DAMAGES OR LOST PROFITS, EVEN IF AII HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. SOME STATES DO NOT ALLOW THE LIMITATION OR EXCLUSION OF LIABILITY FOR INCIDENTAL OR CONSEQUENTIAL DAMAGES, SO THE ABOVE LIMITATION OR EXCLUSION TO INCIDENTAL OR CONSEQUENTIAL DAMAGES MAY NOT APPLY TO LICENSEE.

7. **Export Control.** This product is subject to the jurisdiction of the United States. Licensee may not export or reexport the Licensed Files, without complying with all United States export laws and regulations, including but not limited to (i) obtaining prior authorization from the U.S. Department of Commerce if a validated export license is required, and (ii) obtaining "written assurances" from licensees, if required.

8. **Support and Maintenance.** Except as may be provided in a separate agreement between AII and Licensee, if any, AII is under no obligation to maintain or support the copies of the Licensed Files made and distributed hereunder and AII has no obligation to furnish Licensee with any further assistance, documentation or information of any nature or kind.

9. **Term.** This License Agreement is effective upon Licensee opening this package and shall continue until terminated. Licensee may terminate this License Agreement at any time by returning the Licensed Materials and all copies thereof and extracts therefrom to AII and certifying to AII in writing that all Licensed Materials and all copies thereof and extracts therefrom have been returned or erased by the memory of Licensee's computer or made non-readable. AII may terminate this License Agreement upon the breach by Licensee of any term hereof. Upon such termination by AII, Licensee agrees to return to AII or destroy the Licensed Materials and all copies and portions thereof.

10. **Governing Law.** This License Agreement shall be construed and governed in accordance with the laws of the State of California.

11. **Severability.** Should any term of this License Agreement be declared void or unenforceable by any court of competent jurisdiction, such declaration shall have no effect on the remaining terms herein.

12. **No Waiver.** The failure of either party to enforce any rights granted hereunder or to take action against the other party in the event of any breach hereunder shall not be deemed a waiver by that party as to subsequent enforcement of rights or subsequent actions in the event of future breaches.

13. **Notes to United States Government Users.** Software and documentation are provided with restricted rights. Use, duplication or disclosure by the government is subject to (i) restrictions set forth in GSA ADP Schedule Contract with AII's reseller(s), or (ii) restrictions set forth in subparagraph (c) (1) and (2) of 48 CFR 52.227-19, as applicable.

14.**Third Party Materials.** Licensee is notified that the Licensed Files contain third party software and materials licensed to AII by certain third party licensors. Some third party licensors (e.g., Wind River and their licensors with respect to the Run-Time Module) are third part beneficiaries to this License Agreement with full rights of enforcement. Please refer to the section entitled "Third Party Licenses and Notices" on page A-4 for the third party license and notice terms.

Third Party Licenses and Notices

The licenses and notices related only to such third party software are set forth below:

A. Booting and Debugging Non-Proprietary Software

A small, separate software portion aggregated with the core software in this product and primarily used for initial booting and debugging constitutes non-proprietary software, some of which may be obtained in source code format from AII for a limited period of time. AII will provide a machine-readable copy of the applicable non-proprietary software to any requester for a cost of copying, shipping and handling. This offer will expire 3 years from the date of the first shipment of this product.

B. The OpenLDAP Public License: Version 2.4, 8 December 2000

C. Linux

Linux is written and distributed under the GNU General Public License which means that its source code is freely-distributed and available to the general public.

D. GNU GENERAL PUBLIC LICENSE: Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc. 675 Mass Ave, Cambridge, MA 02139, USA Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation’s software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author’s protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors’ reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone’s free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

GNU GENERAL PUBLIC LICENSE TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0 This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The “Program”, below, refers to any such program or work, and a “work based on the Program” means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term “modification”.) Each licensee is addressed as “you”.

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1 You may copy and distribute verbatim copies of the Program’s source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2 You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

> **a** You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.
>
> **b** You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.
>
> **c** If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it. Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3 You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

> **a** Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

b Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

c Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4 You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5 You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6 Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7 If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8 If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9 The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and “any later version”, you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10 If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11 BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM “AS IS” WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12 IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

Appendix: How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

> <one line to give the program's name and a brief idea of what it does.> Copyright (C)
> 19yy <name of author>
>
> This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.
>
> This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.
>
> You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 675 Mass Ave, Cambridge, MA 02139, USA.

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

> Gnomovision version 69, Copyright (C) 19yy name of author Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type 'show w'. This is free software, and you are welcome to redistribute it under certain conditions; type 'show c' for details.

The hypothetical commands 'show w' and 'show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than 'show w' and 'show c'; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

> Yoyodyne, Inc., hereby disclaims all copyright interest in the program 'Gnomovision' (which makes passes at compilers) written by James Hacker.
>
> <signature of Ty Coon>, 1 April 1989
> Ty Coon, President of Vice

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Library General Public License instead of this License.

> URLWatch:
> For notice when this page changes, fill in your email address.

Maintained by: Webmaster, Linux Online Inc.
Last modified: 09-Aug-2000 02:03AM.
Views since 16-Aug-2000: 177203.

Material copyright Linux Online Inc.
Design and compilation copyright (c)1994-2002 Linux Online Inc.
Linux is a registered trademark of Linus Torvalds
Tux the Penguin, featured in our logo, was created by Larry Ewing
Consult our privacy statement

URLWatch provided by URLWatch Services.
All rights reserved.

E. University of California

Provided with this product is certain TCP input and Telnet client software developed by the University of California, Berkeley.

F. Carnegie-Mellon University

Provided with this product is certain BOOTP Relay software developed by Carnegie-Mellon University.

G. Random.c

PR 30872 B Kesner created May 5 2000
PR 30872 B Kesner June 16 2000 moved batch_entropy_process to own task iWhirlpool to make code more efficient

random.c -- A strong random number generator

Version 1.89, last modified 19-Sep-99

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, and the entire permission notice in its entirety, including the disclaimer of warranties.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

3. The name of the author may not be used to endorse or promote products derived from this software without specific prior written permission. ALTERNATIVELY, this product may be distributed under the terms of the GNU Public License, in which case the provisions of the GPL are required INSTEAD OF the above restrictions. (This clause is necessary due to a potential bad interaction between the GPL and the restrictions contained in a BSD-style copyright.)

THIS SOFTWARE IS PROVIDED “AS IS” AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE, ALL OF WHICH ARE HEREBY DISCLAIMED. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF NOT ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

H. Apptitude, Inc.

Provided with this product is certain network monitoring software ("MeterWorks/RMON") licensed from Apptitude, Inc., whose copyright notice is as follows: Copyright (C) 1997-1999 by Apptitude, Inc. All Rights Reserved. Licensee is notified that Apptitude, Inc. (formerly, Technically Elite, Inc.), a California corporation with principal offices at 6330 San Ignacio Avenue, San Jose, California, is a third party beneficiary to the Software License Agreement. The provisions of the Software License Agreement as applied to MeterWorks/RMON are made expressly for the benefit of Apptitude, Inc., and are enforceable by Apptitude, Inc. in addition to AII. IN NO EVENT SHALL APPTITUDE, INC. OR ITS SUPPLIERS BE LIABLE FOR ANY DAMAGES, INCLUDING COSTS OF PROCUREMENT OF SUBSTITUTE PRODUCTS OR SERVICES, LOST PROFITS, OR ANY SPECIAL, INDIRECT, CONSEQUENTIAL OR INCIDENTAL DAMAGES, HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, ARISING IN ANY WAY OUT OF THIS AGREEMENT.

I. Agranat

Provided with this product is certain web server software ("EMWEB PRODUCT") licensed from Agranat Systems, Inc. ("Agranat"). Agranat has granted to AII certain warranties of performance, which warranties [or portion thereof] AII now extends to Licensee. IN NO EVENT, HOWEVER, SHALL AGRANAT BE LIABLE TO LICENSEE FOR ANY INDIRECT, SPECIAL, OR CONSEQUENTIAL DAMAGES OF LICENSEE OR A THIRD PARTY AGAINST LICENSEE ARISING OUT OF, OR IN CONNECTION WITH, THIS DISTRIBUTION OF EMWEB PRODUCT TO LICENSEE. In case of any termination of the Software License Agreement between AII and Licensee, Licensee shall immediately return the EMWEB Product and any back-up copy to AII, and will certify to AII in writing that all EMWEB Product components and any copies of the software have been returned or erased by the memory of Licensee's computer or made non-readable.

J. RSA Security Inc.

Provided with this product is certain security software ("RSA Software") licensed from RSA Security Inc. RSA SECURITY INC. PROVIDES RSA SOFTWARE "AS IS" WITHOUT ANY WARRANTY WHATSOEVER. RSA SECURITY INC. DISCLAIMS ALL WARRANTIES, EXPRESS, IMPLIED OR STATUTORY, AS TO ANY MATTER WHATSOEVER INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NON-INFRINGEMENT OF THIRD PARTY RIGHTS.

K. Sun Microsystems, Inc.

This product contains Coronado ASIC, which includes a component derived from designs licensed from Sun Microsystems, Inc.

L. Wind River Systems, Inc.

Provided with this product is certain software ("Run-Time Module") licensed from Wind River Systems, Inc. Licensee is prohibited from: (i) copying the Run-Time Module, except for archive purposes consistent with Licensee's archive procedures; (ii) transferring the Run-Time Module to a third party apart from the product; (iii) modifying, decompiling, disassembling, reverse engineering or otherwise attempting to derive the source code of the Run-Time Module; (iv) exporting the Run-Time Module or underlying technology in contravention of applicable U.S. and foreign export laws and regulations; and (v) using the Run-Time Module other than in connection with operation of the product. In addition, please be advised that: (i) the Run-Time Module is licensed, not sold and that AII and its licensors retain ownership of all copies of the Run-Time Module; (ii) WIND RIVER DISCLAIMS ALL IMPLIED WARRANTIES, INCLUDING WITHOUT LIMITATION THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, (iii) The SOFTWARE LICENSE AGREEMENT EXCLUDES LIABILITY FOR ANY SPECIAL, INDIRECT, PUNITIVE, INCIDENTAL AND CONSEQUENTIAL DAMAGES; and (iv) any further distribution of the Run-Time Module shall be subject to the same restrictions set forth herein. With respect to the Run-Time Module, Wind River and its licensors are third party beneficiaries of the License Agreement and the provisions related to the Run-Time Module are made expressly for the benefit of, and are enforceable by, Wind River and its licensors.

M.Network Time Protocol Version 4

The following copyright notice applies to all files collectively called the Network Time Protocol Version 4 Distribution. Unless specifically declared otherwise in an individual file, this notice applies as if the text was explicitly included in the file.

```
***********************************************************************
*                                                                     *
* Copyright (c) David L. Mills 1992-2003                              *
*                                                                     *
* Permission to use, copy, modify, and distribute this software and   *
* its documentation for any purpose and without fee is hereby         *
* granted, provided that the above copyright notice appears in all    *
* copies and that both the copyright notice and this permission       *
* notice appear in supporting documentation, and that the name        *
* University of Delaware not be used in advertising or publicity      *
* pertaining to distribution of the software without specific,        *
* written prior permission. The University of Delaware makes no       *
* representations about the suitability this software for any        *
* purpose. It is provided "as is" without express or implied          *
* warranty.                                                           *
*                                                                     *
*************************************************************************
```

Index